



智能制造与装备制造业转型升级丛书



人工心理与数字人技术

BIPED WALKING ROBOT

双足步行机器人

解仑 王志良 李敏嘉 / 编著



双足步行机器人代表着机器人研究应用领域的水平

本书包括双足步行机器人的发展和国内外动态、基础理论与方法、相关技术实验平台、行为生成技术、情感化步态、12个典型实验

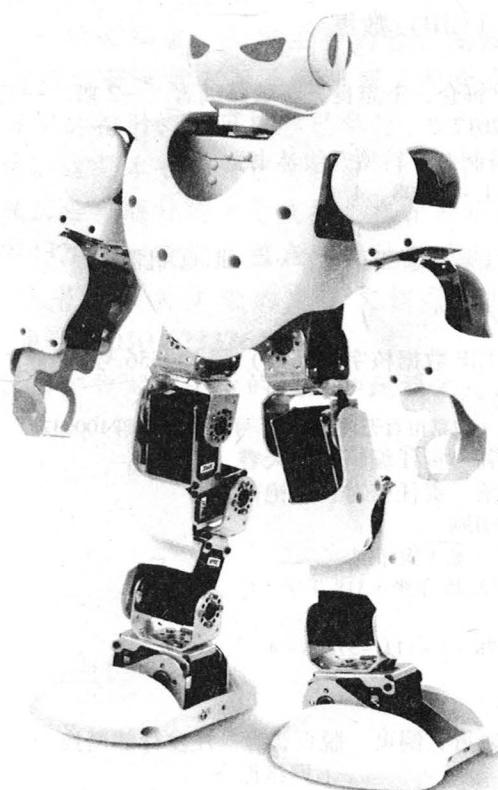


智能制造与装备制造业转型升级丛书

BIPED WALKING ROBOT

双足步行机器人

解仑 王志良 李敏嘉 / 编著



本书在介绍分析双足步行机器人的发展动态和相关理论、技术的基础上，着重通过实验操作的方法向读者讲述双足步行机器人的制作技术。在介绍双足步行机器人概况、理论与方法、相关技术、实验平台、动作生成器、情感化步态等内容之后，针对双足步行机器人设计了 12 个标准实验，这些实验都可以在基于作者自主研究开发的双足步行机器人平台上实际运行，并给出了所有的参考答案，以飨读者。

本书可以作为大学本科及专科学生、职业技术类学生和中学师生的机器人实验教学参考书，也适宜从事计算机、机械电子工程、自动化、智能科学、数字娱乐的科技人员阅读和研究。

图书在版编目 (CIP) 数据

双足步行机器人 / 解仑, 王志良, 李敏嘉编著. —2 版. —北京：
机械工业出版社, 2017.7
(智能制造与装备制造业转型升级丛书)
ISBN 978 - 7 - 111 - 57208 - 4

I. ①双… II. ①解…②王…③李… III. ①机器人
技术 IV. ①TP24

中国版本图书馆 CIP 数据核字 (2017) 第 146556 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑：罗 莉 责任编辑：翟天睿

责任印制：常天培 责任校对：胡艳萍

北京京丰印刷厂印刷

2017 年 7 月第 2 版 · 第 1 次印刷

184mm × 260mm · 13.25 印张 · 318 千字

0 001—3 000 册

标准书号：ISBN 978 - 7 - 111 - 57208 - 4

定价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务 网络服务

服务咨询热线：010-88361066 机工官网：www.cmpbook.com

读者购书热线：010-68326294 机工官博：weibo.com/cmp1952

010-88379203 金书网：www.golden-book.com

封面无防伪标均为盗版 教育服务网：www.cmpedu.com

前　　言

智能制造与装备制造业转型升级的重要技术基础与应用领域之一是机器人科学与技术，而双足步行机器人又代表着机器人研究应用领域的最高水平，是青少年学习掌握机器人技术的重要途径和手段，主要涉及人工智能、计算机硬件制作与软件控制、先进传感器技术等。如何更好地让青少年学习掌握这门技术，尤其是通过实验制作的方法，由浅入深地向读者传授双足步行机器人技术，是我们撰写这本书的最终目的。

本书以大学本科和专科学生、职业技术类学生和中学师生为主要读者对象，详细介绍双足步行机器人的制作技术与方法。全书共分为 7 章。第 1 章主要介绍双足步行机器人的发展过程和国内外研究动态；第 2 章介绍双足步行机器人的基础理论和方法；第 3 章介绍双足步行机器人的相关技术；第 4 章介绍双足步行机器人的实验平台；第 5 章介绍双足步行机器人的行为生成技术；第 6 章介绍双足机器人情感化步态；第 7 章介绍双足步行机器人的 12 个典型实验，并给出其参考答案。

本书由解仑、王志良、李敏嘉编著，制定主要内容、划分章节和统稿。其中第 7 章的 12 个实验都可以在基于作者研究开发的双足步行机器人平台上进行实际运行，读者只要具备我们的这种环境，按照书中的指引，就可以自己学习、动手做实验，最终真正掌握双足步行机器人的制作技术与方法。这是本书的特色所在。

本书的出版得到了机械工业出版社的大力支持，在此表示诚挚的感谢，同时感谢国家自然科学基金（61672093，61432004）和国家重点研发计划重点专项课题（2016YFB1001404）的支持。作者可以免费提供书中相关实验的电子文档及程序。作者的电子信箱 E-mail：jiqiren_ustb@sohu. com，电话：010-62332641。

由于作者的水平有限，书中肯定有不少的缺点和疏漏之处，敬请读者批评指正。

目 录

| | |
|---------------------------|-----------|
| 前言 | |
| 第1章 绪论 | 1 |
| 1.1 机器人概述 | 1 |
| 1.1.1 机器人发展史 | 1 |
| 1.1.2 机器人的分类 | 2 |
| 1.1.3 机器人的定义与三定律 | 3 |
| 1.1.4 机器人技术发展动向 | 4 |
| 1.1.5 类人机器人技术 | 4 |
| 1.1.6 机器人系统理论 | 7 |
| 1.1.6.1 机械系统 | 7 |
| 1.1.6.2 人工心理 | 7 |
| 1.1.6.3 智能控制 | 8 |
| 1.1.6.4 人工生命 | 9 |
| 1.1.7 双足步行机器人 | 9 |
| 1.2 各国研究状况 | 10 |
| 1.2.1 日本和韩国 | 10 |
| 1.2.1.1 日本本田公司 | 10 |
| 1.2.1.2 日本索尼公司 | 12 |
| 1.2.1.3 其他 | 14 |
| 1.2.1.4 日本机器人研究主要代表人物 | 14 |
| 1.2.1.5 韩国的机器人发展 | 16 |
| 1.2.2 美国和欧洲 | 16 |
| 1.3 国内研究状况 | 19 |
| 1.4 其他相关技术 | 22 |
| 1.5 相关参考资料 | 23 |
| 1.5.1 学术会议和期刊 | 23 |
| 1.5.2 网站 | 23 |
| 1.6 小结 | 24 |
| 参考文献 | 24 |
| 第2章 双足步行机器人相关理论与方法 | 25 |
| 2.1 概述 | 25 |
| 2.2 双足步行机器人相关理论与方法的形成及发展 | 26 |
| 2.2.1 稳定性与控制方法 | 26 |
| 2.2.2 双足步行机器人行为发生方法 | 27 |
| 2.2.3 人机接口方法 | 27 |
| 2.2.3.1 人机接口方法的改进 | 30 |
| 2.2.3.2 人机接口方法的发展趋势 | 31 |
| 2.3 理论难点 | 33 |
| 2.3.1 人工智能理论 | 33 |
| 2.3.2 人工心理与人工情感 | 34 |
| 2.3.3 遗传算法的应用 | 34 |
| 2.3.4 基于仿生学原理的步态控制 | 35 |
| 2.3.5 动力学模型 | 36 |
| 2.3.5.1 动力学分析 | 37 |
| 2.3.5.2 ZMP 稳定性判据 | 37 |
| 2.3.6 人机交互的实现方法 | 39 |
| 2.4 小结 | 40 |
| 参考文献 | 40 |
| 第3章 双足步行机器人的相关技术 | 42 |
| 3.1 动作自动生成——行为数据库技术 | 42 |
| 3.2 3D 虚拟仿真技术 | 42 |
| 3.3 双足步行机器人建模的方法与技术 | 44 |
| 3.3.1 ADAMS 建模方法 | 44 |
| 3.3.2 模型的数据转换 | 44 |
| 3.4 多智能体技术 | 45 |
| 3.4.1 智能体通信技术 | 46 |
| 3.4.1.1 通信方式 | 46 |
| 3.4.1.2 通信语言 | 47 |
| 3.4.1.3 通信模型 | 47 |
| 3.4.1.4 通信服务器 | 47 |
| 3.4.1.5 通信服务 | 48 |
| 3.4.2 多智能体机器人系统的数据通信与协调控制 | 48 |
| 3.5 小结 | 49 |
| 参考文献 | 49 |
| 第4章 双足步行机器人实验平台 | 51 |
| 4.1 双足步行机器人控制系统 | 52 |
| 4.2 双足步行机器人的上位机软件 | 53 |
| 4.2.1 初始化设定模块 | 53 |
| 4.2.2 机器人的运行模式与通信协议 | 55 |
| 4.2.3 双足步行机器人控制端软件 | 56 |

| | | | |
|--------------------------------------|----|--|-----|
| 4.2.4 多机器人控制 | 59 | 5.5 小结 | 84 |
| 4.3 总体设计架构 | 61 | 参考文献 | 84 |
| 4.4 小结 | 61 | 第6章 双足机器人情感化步态 | 86 |
| 参考文献 | 62 | 6.1 引言 | 86 |
| 第5章 双足步行机器人的行为生成 技术 | 63 | 6.2 情感模型相关理论 | 86 |
| 5.1 多智能体控制的实现 | 63 | 6.3 基于欧式空间的人工情感模型及 Ekman 情感理论 | 88 |
| 5.1.1 多智能体的特点 | 63 | 6.4 双足机器人步态情感模型的建立 | 89 |
| 5.1.2 多机器人动作协调设计及转化 程序的实现 | 64 | 6.5 基于情感的双足运动系统 | 92 |
| 5.1.3 “千手观音”机器人多智能体 控制的实现 | 67 | 参考文献 | 95 |
| 5.2 双足步行机器人的情感模型、行为决 策及情绪熵 | 71 | 第7章 双足步行机器人技术的课 程实验 | 96 |
| 5.2.1 隐马尔可夫模型 | 71 | 实验一 典型 PIC 单片机控制芯片的编程 实验 | 96 |
| 5.2.2 情感模型及行为决策的设计 | 72 | 实验二 常用传感器实验 | 111 |
| 5.2.3 机器人情感转移控制的具体 实现 | 74 | 实验三 舵机控制实验 | 118 |
| 5.2.4 情绪熵的选取 | 75 | 实验四 机械结构设计与认识实验 | 127 |
| 5.3 双足步行机器人的语音控制 | 77 | 实验五 简单动作调试实验 | 133 |
| 5.3.1 语音在双足步行机器人控制平台 上的实现 | 77 | 实验六 动作数据编程实验 | 142 |
| 5.3.2 语音数据库的设计 | 80 | 实验七 上位机编程实验 | 150 |
| 5.4 双足步行机器人动作的设计 | 82 | 实验八 双足步行机器人串行口通信 实验 | 157 |
| 5.4.1 基于情感方面的动作设计 | 82 | 实验九 ZMP 实验 | 168 |
| 5.4.2 基于性格方面的动作设计 | 83 | 实验十 多智能体协调控制 | 177 |

第1章 絮 论

1.1 机器人概述

1.1.1 机器人发展史

机器人的历史源远流长，早在我国的西周时期，一名叫做偃师的能工巧匠就造就了一个能歌善舞的伶人，这是有据可查的第一个“机器人”。在1800年前的汉朝，张衡造出了举世闻名的地动仪和计里鼓车。在三国时期的诸葛亮发明了木牛流马，用来运送粮草。在国外，公元前2世纪亚历山大时期，古希腊人造就出了“自动机”——以空气、水、蒸汽压力为动力的会动的雕像。这些都可以看做是广义上的机器人。

美国的德沃尔（George G. Devol）设想了一种可控制的机械手，他首先突破了对机器人的传统观点，提出机器人并不一定必须像人，但是必须能做一些人的工作。1954年，他依据这一想法设计制作了世界上第一台机器人实验装置，发表了《适用于重复作业的通用性工业机器人》一文，并获得了美国专利。

德沃尔将遥控操纵器的关节型连杆机构与数控机床的伺服轴连接在一起，预定的机械手动作一经编程输入后，机械等就可以离开人的辅助而独立运行。这种机器人也可以接受示教而完成各种简单任务。

1959年，英格伯格和德沃尔联手制造出第一台工业机器人。由英格伯格负责设计机器人的“手”“脚”“身体”，即机器人的机械部分和完成操作部分；由德沃尔设计机器人的“头脑”“神经系统”“肌肉系统”，即机器人的控制装置和驱动装置。它成为世界上第一台真正的实用工业机器人。

英格伯格在大学攻读伺服理论，这是一种研究运动机构如何才能更好地跟踪控制信号的理论。德沃尔曾于1946年发明了一种系统，可以“重演”所记录的机器的运动。1954年，德沃尔又获得可编程机械手专利，这种机械手臂按程序进行工作，可以根据不同的工作需要编制不同的程序，因此具有通用性和灵活性，英格伯格和德沃尔都在研究机器人，他们认为汽车工业最适于用机器人干活，因为汽车工业是用重型机器进行工作的，生产过程较为固定。

这种机器人外形有点像坦克炮塔，基座上有一个大机械臂，大臂可绕轴在基座上转动，大臂上又伸出一个小机械臂，它相对大臂可以伸出或缩回。小臂顶有一个腕子，可绕小臂转动，进行俯仰和侧摇。腕子前头是手，即操作器。这个机器人的功能和人手臂功能相似，如图1-1所示。

这是世界上第一台真正的实用工业机器人。此后英格伯格和德沃尔成立了“Unimation”公司，兴办了世界上第一家机器人制造工厂。第一批工业机器人被称为“尤尼梅特（UNIMATE）”，意思是“万能自动”。他们因此被称为机器人之父。1962年美国机械与铸造公司

也制造出工业机器人，称为“沃尔萨特兰（VERSTRAN）”，意思是“万能搬动”。“尤尼梅特”和“沃尔萨特兰”就成为世界上最早的、至今仍在使用的工业机器人。

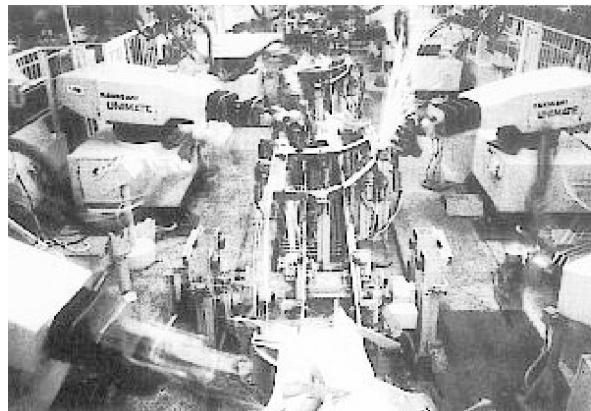


图 1-1 世界上第一台工业机器人“尤尼梅特”正在生产线上工作

英格伯格和德沃尔制造的工业机器人是第一代机器人，属于示教再现型，即人手把着机械手，把应当完成的任务做一遍，或者人用“示教控制盒”发出指令，让机器人的机械手臂运动，一步步完成它应当完成的各个动作。

1967 年日本川崎重工业公司从美国购买了机器人的生产许可证，日本从此开始了对机器人制造和开发的热潮。

20 世纪 80 年代，计算机技术推动机器人技术的发展达到了新的水平。上到宇宙飞船，下至海洋开发都采用了机器人作业。机器人技术已成为高科技应用领域中的重要组成部分。

第二代是有感觉的机器人，它们对外界环境有一定感知能力，并具有听觉、视觉、触觉等功能。机器人工作时，根据感觉器官（传感器）获得的信息，灵活调整自己的工作状态，保证在适应环境的情况下完成工作。如：有触觉的机械手可轻松自如地抓取鸡蛋，具有嗅觉的机器人能分辨出不同饮料和酒类。

第三代机器人是智能机器人，它不仅具有感觉能力，而且还具有独立判断和行动的能力，并具有记忆、推理和决策的能力，因而能够完成更加复杂的动作。中央计算机控制手臂和行走装置，使机器人的手完成作业，脚完成移动，机器人能够用自然语言与人对话。智能机器人的“智能”特征就在于它具有与外部世界——对象、环境和人相适应、相协调的工作机能。从控制方式看，智能机器人不同于工业机器人的“示教、再现”，不同于遥控机器人的“主—从操纵”，而是以一种“认知—适应”的方式自律地进行操作。

智能机器人在发生故障时，通过自诊断装置能自诊断出故障部位，并能自修复。今天，智能机器人的应用范围大大地扩展了，除工农业生产外，机器人还应用到各行各业，机器人已具备了人类的特点。机器人向着智能化、拟人化方向发展的道路是没有止境的。

1.1.2 机器人的分类

机器人按不同的分类方式可分为很多种，如按应用环境可分为工业机器人和特种机器人；按功能可分为操作型机器人、程控型机器人、示教再现型机器人、数控型机器人、感觉控制型机器人、适应控制型机器人、学习控制型机器人和智能机器人等。

关于机器人如何分类，国际上没有制定统一的标准，有的按功能分，有的按负载重量分，有的按控制方式分，有的按自由度分，有的按结构分，有的按应用环境分。我国的机器人专家从应用环境出发，将机器人分为两大类：工业机器人和特种机器人。所谓工业机器人就是面向工业领域的多关节机械手或多自由度机器人。而特种机器人则是除工业机器人之外的、用于非制造业并服务于人类的各种先进机器人。包括服务机器人、水下机器人、娱乐机器人、军用机器人、农业机器人、探索机器人等。在特种机器人中，有些分支发展很快，有独立成体系的趋势，如服务机器人、水下机器人、军用机器人、操作机器人等。目前，国际上的机器人学者，从应用环境出发将机器人也分为两类：制造环境下的工业机器人和非制造环境下的服务性类人机器人，这和我国的分类是一致的。

1. 按照机器人从低级到高级的发展程度分类

(1) 第一代机器人 第一代机器人主要指只能以“示教—再现”方式工作的工业机器人。目前国际上商品化、实用化的机器人大都属于第一代机器人。

(2) 第二代机器人 第二代机器人是智能机器人。它具有多种感知功能，可进行复杂的逻辑思维、判断决策，在作业环境中独立行动。

(3) 第三代机器人 第三代机器人是未来机器人。

2. 按照开发内容和目的分类

按照开发内容和目的分类基本上有3类机器人：①工业机器人（Industrial Robot）；②操纵型机器人（Teleoperator Robot）；③智能机器人（Intelligent Robot）。

3. 按照机器人的结构形式分类

机器人可分为关节型机器人和非关节型机器人。其中关节型机器人的机械本体部分一般为由各种关节串接起若干连杆组成的开链式机构。

1.1.3 机器人的定义与三定律

在科学界，科学家会给每一个科技术语一个明确的定义，机器人问世已有几十年了，但是对于机器人的定义仍然是仁者见仁，智者见智，没有一个统一的意见。原因之一是机器人还在发展，新的机型、新的功能不断涌现。根本原因主要是因为机器人涉及了人的概念，成为一个难以回答的哲学问题。

1886年法国作家利尔亚在他的小说《未来的夏娃》中将外表像人的机器起名为“安德罗丁”。1920年捷克作家卡雷尔·卡佩克发表了科幻剧本《罗萨姆德万能机器人》。剧本中，卡佩克把捷克语“Robota”写成了“Robot”，该剧预告了机器人的发展对人类社会的悲剧性影响，引起了大家的广泛关注，被当成了机器人一词的起源。

在1967年日本召开了第一届机器人学术会议上，人们提出了两个有代表性的定义。一个是森政弘与合田周平提出的：“机器人是一种具有移动性、个体性、智能性、通用性、半机器半人性、自动性、奴隶性等7个特征的柔性机器”。从这一定义出发，森政弘又提出了自动性、智能性、个体性、半机器半人性、作业性、通用性、信息性、柔性、有限性、移动性等10个特性来表示机器人的形象；另一个是加藤一郎提出的具有如下3个条件的机器称为机器人：

具有脑、手、脚等三要素的个体；

具有非接触传感器（用眼、耳接受远方信息）和接触传感器；

具有平衡觉和固有觉得传感器。

1987 年国际标准化组织 ISO 对工业机器人进行了定义：“工业机器人是一种具有自动控制的操作和移动功能，能完成各种作业的可编程操作机^[7]。”

我国科学家对机器人的定义是：“机器人是一种自动化的机器，所不同的是这种机器具备一些与人或生物相似的智能能力，如感知能力、规划能力、动作能力和协同能力，是一种具有高度灵活性的自动化机器。”在研究和开发未知以及不确定环境下作业的机器人的过程中，人们逐步认识到机器人技术的本质是感知、决策、行动和交互技术的结合。

现代的机器人技术在不断地发展着，随着机器人的进化和人工智能的进步，这些定义都会得到不断的完善和发展，甚至有可能重新对机器人进行崭新的定义。

美国的艾萨克·阿西莫夫是一位化学家和有着世界声望的科普作家，他对科幻小说的最大贡献之一是“发明”了“机器人三定律”。三定律从逻辑学的角度对科幻小说中恣意妄为的机器人进行了行为规范，对今天的机器人研制也有着重要的指导意义。

机器人学三定律：

第一定律——机器人不得伤害人，也不得见人受到伤害而袖手旁观。

第二定律——机器人应服从人的一切命令，但不得违反第一定律。

第三定律——机器人应保护自身的安全，但不得违反第一、第二定律。

1.1.4 机器人技术发展动向

当前和今后的机器人技术正逐渐向着具有行走能力、对环境的自主性强、具有多种感觉（比如可以感觉形状、重量、硬度、温度和湿度等）能力的方向发展。

机器人也正在逐渐具有智能。美国贝尔科尔公司已成功地将神经网络装配在芯片上，其智能分析速度比普通计算机要快数千倍，能更好地完成识别语言和图像处理等工作。

目前，对机器人技术的发展有最重要影响的国家是美国和日本。美国在机器人技术的综合性水平上仍处于领先地位，日本生产的机器人数量和种类则居世界首位。

我国发展机器人技术起步于 20 世纪 70 年代末。1995 年 9 月，6000m 水下机器人试验成功。近年来，在步行机器人、精密装配机器人及多自由度关节型机器人研制等前沿领域内逐步缩短了与世界水平的差距。

机器人技术经数十年的发展，现已形成了一门综合性学科——机器人学（Robotics）。机器人学包括基础研究和应用研究两方面内容，主要研究领域有：①机械手设计；②机器人运动学、动力学和控制；③轨迹设计和规划；④机器人传感器；⑤机器人视觉；⑥机器人控制语言和离线编程；⑦机器人本体结构系统；⑧机器智能等。

1.1.5 类人机器人技术

现阶段，机器人的研究应用领域不断拓宽，其中类人机器人（也叫仿人机器人）的研究和应用尤其受到普遍关注，并成为智能机器人领域中最活跃的研究热点之一。

最早系统地研究人类和动物运动原理的是美国摄影师爱德华得·麦布里奇（Eadweard Muybridge），他发明了电影用的独特摄像机，即一组电动式触发照相机，并在 1877 年成功地拍摄了许多四足动物步行和奔跑的连续照片。后来，这种采用摄像机的方法又被乔治·德米尼（Georges Demeny）用来研究人类的步行运动。

真正全面、系统地开展类人机器人的研究是始于 20 世纪 60 年代。而类人机器人的重要基础研究内容，就是双足步行机器人技术。迄今，不仅形成了双足步行机器人一整套较为完善的理论体系，而且在一些国家，如日本、美国和俄罗斯等都已研制成功了能静态或动态步行的双足步行机器人样机。

早在 20 世纪 50 年代中期，美国通用电气公司就制造了一台名为“Hardiman”的步行车，但当时的驱动和伺服控制技术显然还不足以使 Hardiman 进入实用化阶段。

1968 年，英国的 Mosher R 试制了一台名为“Rig”的操纵型双足步行机器人，它只有踝和髋两个关节，操作者靠力反馈感觉来保持机器人平衡，这种主从式的机械装置可算是双足步行机构的雏形。

真正的双足步行机器人是 I. Kato 在 1971 年试制的 Wap3，它最大步幅为 15mm，周期为 45s，Wap3 的研制成功，揭开了双足步行机器人的研究序幕。

1984 年，日本的加藤实验室又推出了 WL-9DR 双足步行机器人。1984 年，加藤在以前的研究基础上采用了踝关节力矩控制，使 WL-10RD 双足步行机器人实现了平稳的动态步行。1986 年，加藤又推出了具有 8 个自由度的 WL-12R 双足步行机器人。

1971~1986 年间，牛津大学的 Witt 等人曾制造和完善了一个双足步行机器人，在平地上走得非常好，步速为 0.23m/s，功率消耗约为 4W。

日本的 J. Furusho 研制了两个系列的能够动态步行的双足步行机构，从 1981 年开始，先后研制了 Kenkyaku-1、Kenkyaku-2、BLR-G1 和 BLR-G2 双足步行机器人。

日本的 Kajita 是日本另一步行机器人研究者。1990 年，他研制成功了一台五连杆平面型双足步行机器人，实现了在不平地面上的稳定动态步行。

1986 年，美籍华人郑元芳博士分别研制成功 SD-1 双足步行机器人和 SD-2 型双足步行机器人，成功地实现了平地上的前进、后退以及左、右侧行和动态步行。1990 年，他首次提出了使双足步行机器人能够走斜坡的控制方案，并应用于 SD-2 双足步行机器人中获得成功实现。

1985 年，美国的 Hodgins 和 Raibert 等人研制了一个用来进行奔跑运动和表演体操动作的平面型双足步行机器人。1986 年，他们用这个机器人进行奔跑实验，着重研究奔跑过程中出现的弹射飞行状态。1988 年和 1990 年，他们又用这个机器人进行了翻筋斗动作实验。Hodgins 和 Raibert 研究这两种运动是因为它们含有丰富的动力学内容，尤其是两者都具有弹射飞行状态。

1986 年，日本本田公司制定了第一个研制拟人机器人的计划，并于 1996 年 11 月展示了一个有两腿两臂的拟人机器人 P2。本田公司在 1997 年 10 月又推出了世界领先的 P3 机器人，如图 1-2 所示。

我国从 20 世纪 80 年代中期才开始研究类人机器人，主要研究单位是国防科技大学和哈尔滨工业大学。

哈尔滨工业大学于 1989 年研制了一台类人机器人，重量为 70kg，高度为 110cm，

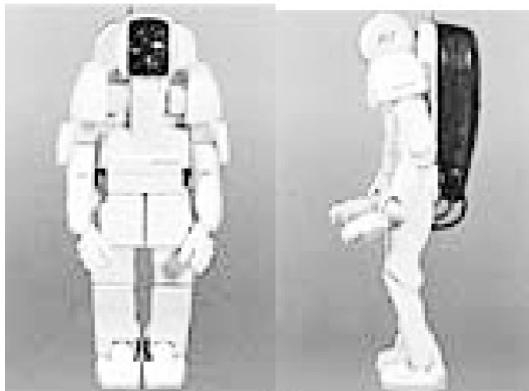


图 1-2 本田 P3 双足步行机器人外观

有 10 个自由度，可以实现平地上的前进、左右侧行及上下楼梯的运动，步幅可达 45cm，步速 10 步/s，为静态步行。

国防科技大学在 1988 年成功研制了一台 KDW-I 平面型、6 自由度的双足步行机器人，能实现前进、后退和上下楼梯。1989 年，他们又成功研制了一台 KDW-II 空间运动型的双足步行机器人，实现了准动态步行，1995 年实现了动态步行。2000 年 11 月 29 日，他们成功研制出新型双足步行机器人。不仅能平地静态步行，而且能快速自如地动态步行；既能在已知环境中步行，也可在小偏差、不确定环境下行走，实现了多项关键技术突破，如图 1-3 所示。

世界著名机器人学专家、日本早稻田大学的加藤一郎教授^[1]说过：“机器人应当具有的最大特征之一是步行功能。”步行是人与大多数生物所具有的移动方式，其形式主要有双足步行、四足步行和六足步行。其中双足步行是步行方式中自动化程度最高、最为复杂的系统，同时双足步行机器人也有着其他机器人不可比拟的优点：具有适应各种地面和较强的逾越障碍的能力；步行机器人的能耗通常低于轮式和履带式机器人，即能耗较小。随着对双足步行机器人的研究不断深入，无论是影视、科幻作品还是人们对机器人的第一意识，都把像人一样的机器人作为机器人研究的最高境界。机器人的研究者也一直把实现机器人的拟人行为作为梦寐以求的目标。拟人机器人是一种智能的、机动的、能满足用户多种需求的新型机器人，这种机器人具有人的外形和基本功能，易于与人共处，易于适应多变的活动环境。20 世纪 90 年代前后，双足机器人从一般性的拟人腿部行走上升到全方位的拟人机器人研究。拟人机器人除了腿部的行走功能外，还包括手、腰和头的功能，自由度比双足步行机器人成倍地增加，与此同时也带来了控制规划、动力学、运动学上更为复杂的问题。此外，还有电荷耦合器件（CCD）图像处理、语音处理以及一系列传感信号的处理。拟人机器人相对于双足机器人的研究，更为类似人类。拟人机器人是由仿生学、机械工程学和控制工程学等多学科相互融合而成的综合性学科。

研究拟人机器人的科学意义主要在于：

第一，拟人机器人的研究模型为控制理论应用及动力学问题的研究提供了广阔的天地。拟人机器人是一个多变量、强耦合、非线性和变结构的复杂动力学系统，其变姿态结构的小稳定性及产生稳定步行运动所需要解决的动态平衡问题，对于控制理论及动力学问题的研究来说，具有很大挑战。

第二，拟人机器人作为步行机器人的一种形式，是提高机器人机动性和节省能源的一条重要途径。研究拟人机器人的双足步行运动，揭示了双足类步行运动的机理及控制规律，研制一种连续稳定步行的拟人步行机器人，可为机器人操作提供灵活的操作平台，使其能够在恶劣的条件下工作。因此拟人机器人有着潜在的广阔应用前景。

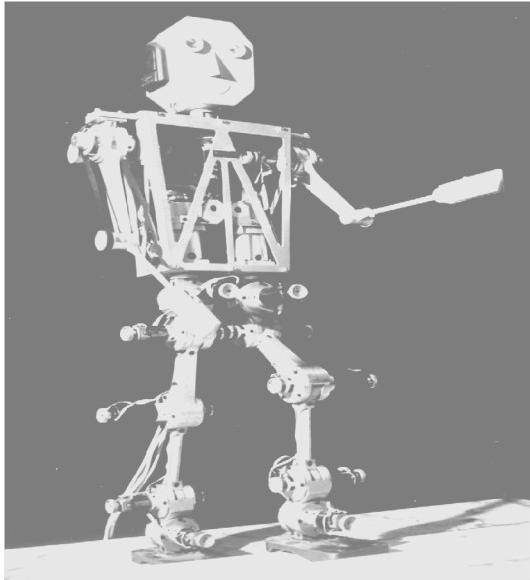


图 1-3 国防科技大学研制的双足步行机器人

拟人机器人研究集机械、电子、计算机、材料、传感器、控制技术、通信、人工智能、人工心理等多门科学于一体，代表着一个国家的高科技发展水平。从机器人技术和人工智能的研究现状来看，要完全实现高智能、高灵活性的拟人机器人还有很长的路要走，而且人类对自身也没有彻底了解，这些都限制了拟人机器人的发展。

1.1.6 机器人系统理论

1.1.6.1 机械系统

机械系统是由机械元素组成的，它可以是自然的，也可以是人造的。后者是我们要研究的对象。机器人的机械系统由以下几个子系统组成：①机械子系统，由刚体和弹性体组成；②传感系统；③执行系统；④控制器；⑤信息处理。此外，子系统之间的通信是通过接口进行的，接口的基本功能是把从一个部分传到另一个部分的信息解码。图 1-4 是典型机械系统的框图，系统的输入是事先确定的任务，它是由实时或离线给定的。前者在本质上看成是智能的，后者是可编程序的机械。机器人的机械系统输出是通过传感器监测的实际任务（Actual Task）。传感器以反馈信号的形式传递作业信息并与事先设定的动作相比较，事先设定的任务与执行动作间的误差反馈给控制器，然后合成必要的校正信号。在机器人控制系统中，通过人来构成闭环控制的称为遥控机器人（Telemanipulator）。遥控机器人是人借助于复杂的传感器和显示装置进行控制的机械系统，操作者也就成了图 1-4 所示框图中的一个中心单元，根据显示的信息，操作者对校正信号调整，以完成所需要完成的动作。

我们必须明确，研究机器人机械系统，需要建立起一套运动的表示方法。我们用独立的杆件运动组合成机械系统，把这些杆件看成刚体。一般刚体的运动由平动和转动组成，通过一个坐标系和矢量代数计算距离和角度，并在此坐标系下描述矢量，但最终的结果与我们所选择的坐标系无关。最终得到的数学模型就是我们要找的机械系统运动的表示方法。这些数学表示方法一般包括：线性变换、刚体的旋转矩阵、坐标变换和齐次坐标建立、校正方法及各种方法的合成等。以上各种方法的具体概念及实现方法请参见相关文献。

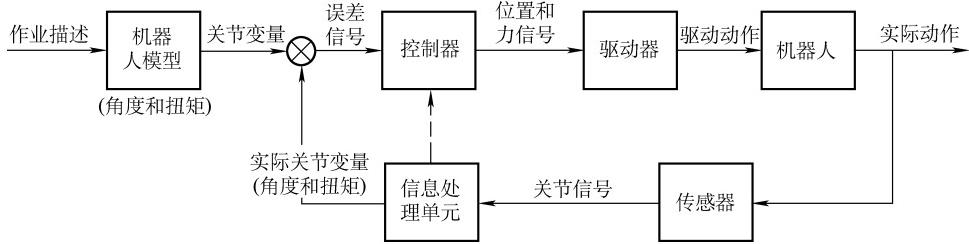


图 1-4 机器人机械系统框图

1.1.6.2 人工心理

用机器模拟人类心理行为（情感等）一直为人们所追求。目前，用机器模拟人类心理行为（情感等）的相关技术研究正在形成一种趋势，尤其以日本最为先进。近几年来，在日本科学技术界流行“感性工学（Kansei Engineering）”的新术语。由于这是一个新的领域，日本所取得的成果都是初步的，有许多问题亟待解决。比如，日本提出的“感性工学”的定义不清楚（现有 10 多种定义），其研究范围只是对准人类心理活动的一个方面——“感性”。王志良教授于 1999 年提出了一个全新的概念“人工心理”，并出现在国家自然科

学基金 2000 年度项目指南自动化学科中。其与人工智能的关系如图 1-5 所示。

人工心理理论，就是利用信息科学手段，对人的心理活动的更全面内容的再一次人工机器（计算机、模型算法）实现——人工心理学。它的应用前景是非常广泛的，如支持开发有情感、意识和智能的机器人；真正意义上的拟人机械研究；使控制理论更接近于人脑的控制模式。我们知道，已有的拟人控制理论主要就是维纳的“反馈”控制论和人工智能；这与人脑的控制模式还有很大差别，因为人脑控制模式是：感知觉 + 情感决定行为；而现有的控制系统决策不考虑也无法考虑情感的因素。人工心理应用的另一大领域是符合人性化的商品设计和市场开发。不夸张地说，人工心理理论是人工智能的高级阶段，是自动化乃至信息科学的全新研究领域，它的研究将会大大促进拟人控制理论、情感机器人、人性化的商品设计和市场开发等方面的进展，为最终营造一个人与人、人与机器和谐的社会环境作出贡献。

1.1.6.3 智能控制

智能控制（Intelligent Control）的概念由傅京逊于 1971 年提出，由美国的萨里迪斯（Saridis）等人在此基础上于 1977 年进一步提出了一种智能机器（Intelligent Machine）的统一结构框架。该结构现已被控制界普遍接受，它具有递阶分散智能与精度的特点，包括组织级（Organization Level）、协调级（Coordination Level）和执行级或控制级（Execution Level），且各层次间遵循精度随智能降低而提高（Increasing Precision with Decreasing Intelligence IPDI）原则。

智能控制是随着被控对象的日益复杂，为适应其对环境依赖的不可侵害性而产生的。作为一门新兴的学科，它融合了包括神经生理学、心理学、运筹学、控制论和计算机科学的多学科思想和技术的成果。而对智能控制的研究，主要体现在对基于知识系统（Knowledge Based System, KBS）、模糊逻辑（Fuzzy Logic）和人工神经网络（Artificial Neural Network）的研究。

机器人学（Robotics）是一门高度综合和交叉的新兴学科，它涉及的领域很多，诸如机械、电气、工艺、力学、传动、控制、通信、决策、生物、伦理等诸多方面。

1) 从控制角度看，其中最主要也是最基本的是机器人的运动学（Kinematics）和动力学（Dynamics）问题及相应的控制策略研究。

机器人运动学包括前向运动学（Forward Kinematics）和逆运动学（Inverse Kinematics），或分别称为运动学的正问题和逆问题。其中，前向运动学研究以机器人的各关节参数（广义坐标）来决定终端操作器（End-effector）的位型（Position and Orientation）；而逆运动学则是研究由期望的终端操作器位型来得到各关节变量的过程。

机器人动力学问题主要包括两大类，即运动分析和力分析。运动分析研究由机器人连杆系的受力情况（外力和关节驱动力）决定各关节运动状况（位置、速度和加速度），进而通过运动学方法来最终获得终端操作器位置的过程和方法；而力分析则是研究由期望的机器人

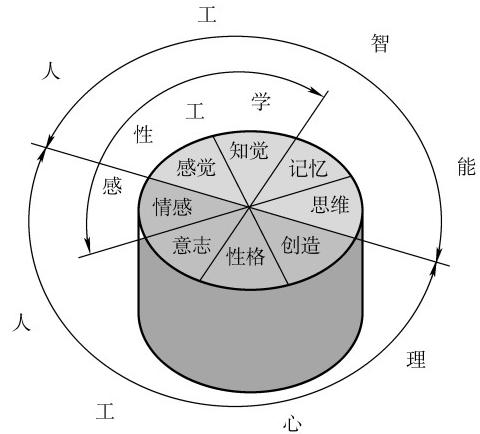


图 1-5 人工心理与人工智能的关系

各连杆的运动（位移、速度和加速度）得到需施加于各关节处的驱动力或力矩的方法和策略，即我们通常所说的控制综合问题。

2) 从智能角度看，随着机器人技术的迅速发展和自动化程序的进一步提高，对机器人的功能也提出了更高的要求，特别是需要各种具有不同程序智能的机器人。智能机器人已在自主系统和柔性加工系统中日益得到广泛的应用。自主机器人能够设定自己的目标，规划并执行自己的动作，使自己不断适应环境的变化。这些机器人智能程度的提高，离不开人工技术的发展，其中包括人工心理、人工智能、情感计算、人工生命等。

1.1.6.4 人工生命

20世纪60年代，人们破译了遗传密码，70年代遗传工程有了重大突破。很自然，生物学研究的下一个目标就是用人工的方法合成生命。然而，以现有生命物质为基础合成生命似乎前景并不乐观。那么，我们有没有其他办法创造生命呢？计算机科学的发展为我们提供了一条新的研究思路，即我们可以尝试在计算机或其他媒质中创造出新形式的生命。这就是80年代末90年代初在国际上兴起的一个新的研究领域——人工生命。人工生命概念一提出，吸引了众多学者参与到这一新兴的研究领域中。

人工生命是一项抽象地提取出控制生物现象的基本动态原理，并且通过物理媒介（如计算机）来模拟生命系统动态发展过程的研究工作。它涉及众多的学科领域，如生物科学、计算机科学、控制科学、系统科学、机器人科学、人工智能等。该项学科的建立一般被认为始于1987年在美国召开的第一次人工生命研讨会。

冯·诺伊曼也是人工科学的先驱。20世纪40年代和50年代，他在数字计算机设计和人工智能领域做了很多开创性的工作。与图灵一样，他也试图用计算的方法揭示出生命最本质的方面。但与图灵关注生物的形态发生不同，他则试图描述生物自我繁殖的逻辑形式。在发现DNA和遗传密码好几年之前，他已认识到，任何自我繁殖系统的遗传物质，无论是自然的还是人工的，都必须具有两个不同的基本功能：一方面，它必须起到计算机程序的作用，是一种在繁衍下一代过程中能够运行的算法；另一方面，它必须起到被动数据的作用，是一个能够复制和传到下一代的描述。为了避免当时电子管计算机技术的限制，他提出了细胞自动机的设想：把一个长方形平面分成很多个网格，每一个格点表示一个细胞或系统的基元，每一个细胞都是一个很简单、很抽象的自动机，每个自动机每次处于一种状态，下一次的状态是由它周围细胞的状态和它自身的状态以及事先定义好的一组简单规则决定的。冯·诺伊曼证明，确实有一种能够自我繁殖的细胞自动机存在，虽然它复杂到了当时的计算机都不能模拟的程度。冯·诺伊曼的这项工作表明：一旦我们把自我繁衍看做是生命独特的特征，机器也能做到这一点。

1.1.7 双足步行机器人

步行是人与大多数动物所具有的移动方式，也是生物界步行方式中自动化程度最高、最为复杂的动作，它的完美实现必然要求机器人在结构设计方面产生巨大的变革和创新，从而有力地推动相关学科的发展。人类一直梦想着创造出和人类构造相似、能与人类合作的拟人机器人。而双足步行机器人是工程上少有的高阶、非线性、非完整约束、强耦合性的多自由度系统，这为机器人运动学、动力学以及控制理论的研究提供了一个理想的实验平台^[2]。

拟人机器人是一种具有人的外形，并能够效仿人体的某些物理功能、感知功能及社交能

力并能承袭人类的部分经验的机器人。当然，拟人机器人的研究目的不是企图制造以假乱真或替代人类的机械，而是要创造一种新型工具，它能在典型的日常环境中和人类交流，在更广泛的环境任务中扩展人类的能力。

双足步行机器人的发展也为人类假肢的发展提供了有力的理论、技术支持，同时也为服务、娱乐机器人的发展开辟了新的领域。随着机器人的工作环境和工作任务的复杂化，双足步行机器人因其体积相对较小、对非结构性环境具有较好的适应性、避障能力强、能耗小、移动盲区很小等优良的移动品质，格外引人注目。拟人机器人不仅具有双腿、双臂、头、眼、颈、腰等物理特征，还能模仿人类的视觉、触觉、语言，甚至情感等功能。这和能在特种环境下工作的服务机器人是有区别的。

另外，双足步行机器人的研究具有十分重大的科研、实用价值和意义，可以推动仿生学、人工智能、计算机图形学、通信等相关学科的发展。拟人机器人的研究是多学科的交叉、综合与提高。机械工程师、电子学家、计算机专家、机器人学家、人工智能专家、心理学家、物理学家、生物学家、认知学家、神经生物学家，甚至哲学家、语言学家和艺术家等都有参与其研究。

1.2 各国研究状况

据韩国一个经常更新的拟人机器人网站统计，目前世界各国约有 85 个大型拟人机器人项目正在进行中，这些还不包括各国单独模拟身体单个部分的项目。科技部 863 自动化办公室的工作人员表示，目前所有的大型拟人机器人项目中，日本占有 37 项，美国占有 10 项，德国占 7 项，韩国占 7 项，英国占 4 项，中国占 5 项，瑞典占 2 项，澳大利亚、泰国、新加坡、保加利亚、伊朗、意大利、奥地利、俄罗斯等国各占有 1 项。从统计数字可看出日本的领先地位及其他各国的竞争实力^[3]。

1.2.1 日本和韩国

在日本所有的研究机构中，早稻田大学、东京工业大学、本田公司、索尼公司等研究机构成为双足步行机器人研究的主力。早稻田大学的双足机器人研究始于 1968 年，该大学的加藤一郎教授所领导的课题组相继研制了 WL 系列、WABIAN 系列双足步行机器人。最新的研究成果为 WL-16 和 WABIAN-2 双足步行机器人，可以在不平的地面上稳定行走^[4]。

在日本，除了各大学都热衷于研究拟人机器人外，公司的研发开始成为主力。许多过去与机器人技术毫不相关的公司都涉足拟人机器人的开发。原因主要有二：一是大公司利用公众对拟人机器人的高度兴趣，通过成功开发拟人机器人，宣传本公司的品牌；二是拟人机器人有一个很大的市场。研究单位或成群的发烧友迫切需要低价的仿人机器人整机平台或套装件。

1.2.1.1 日本本田公司

日本本田公司的仿人机器人工程开始于 1986 年。1986 年本田的 ASIMO 机器人实现了可以两条腿行走，成功地实现了双腿的交替移动。但是，每一步几乎要用 5s 的时间，它只能沿着直线以很慢的速度行走。第一个完全独立，依靠两条腿行走的拟人机器人诞生于 1997 年 9 月，身高：1600mm 体重：130kg。通过改换部件材料，以及将控制系统分散布置，降低了身高，减轻了体重。较小的尺寸更适宜在人们的生活中使用。

经过 10 年的秘密研究，P2 机器人^[8,9]在 1996 年一经问世，便令世人瞠目。P2 机器人是世界上第一个无缆拟人机器人，它能走，能上下台阶，能推小车。P3 是 P2 的改进，减小了尺寸（重量由原来的 210kg 降为 130kg，高度由 1800mm 降为 1600mm，提高了步行能力。2002 年 12 月本田公司又推出了新型双足步行机器人阿西莫 ASIMO，如图 1-6 所示。和“P3”相比，它的体形更小、重量更轻，并采用了新开发的智能实时灵活步行“I-Walkinga (Intelligent Realtime Flexible Walking) 技术”实现更加自由的步行。本田公司这个最新款的 ASIMO 智能拟人机器人主要有以下几个特点：①基于认知技术的先进的交流能力。它能实现移动物体认知、位置/姿态认知、环境认知、声音认知和面部认知。②网络集成能力，它能与用户网络系统相连完成用户指令，还能通过互联网为主人的问题寻找答案。



图 1-6 ASIMO 拟人机器人

2004 年 12 月 15 日本田公开了新一代 ASIMO 双足行走机器人，提高了移动性能，其目标是可在实际环境中快速判断情况、敏捷行动。通过利用上半身弯曲及扭曲的新姿态控制理论，以及新开发的高速响应硬件，加上可与慢跑媲美的动作，实现了时速 3km 行走，正常行走时速为 2.5km/h，能实现跑动时速为 6km/h。其高度为 1.3m，总重为 54kg，具有 34 个自由度，其中下肢 12 个自由度，如图 1-7 所示。它采用与人一样自然行走的新姿态控制技术、自律连续移动技术，以及可顺畅地与人同步动作的技术等。

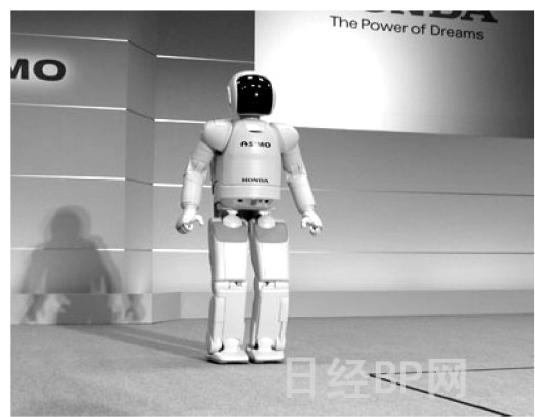
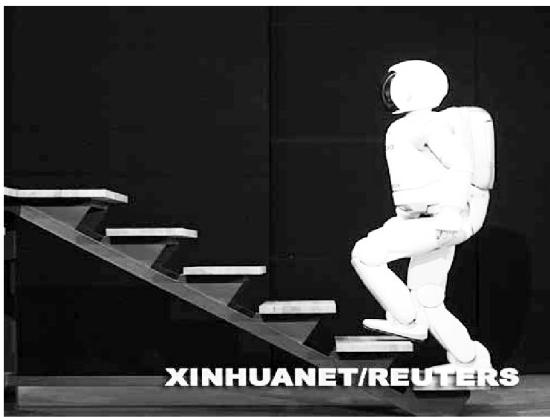


图 1-7 新一代“ASIMO”双足行走机器人

在连续移动方面，它通过由地面传感器获得的周围信息，以及在步行的同时对照事先存入的地图信息，可对路线错误进行修正；当检测到有障碍物时，可凭借自己的判断而绕开；同时，通过头部的视觉传感器及新追加的手腕力觉传感器等，可检测人的动作、接拿物品，以及按照对方的动作节奏握手；可以做出将脚迈向手部被推拉的方向等与人动作同步的动作。

2007 年 9 月 27 日，本田公司主导开发的新版 ASIMO 机器人在西班牙巴塞罗那亮相。经过 7 年研发的新版 ASIMO 机器人体高为 1.2m，体重为 43kg，由移动电话控制，能有以酷似

人类的姿势活动，与以前版本的机器人相比，加强了与人类交流的能力。

1.2.1.2 日本索尼公司

提到拟人机器人，我们就不得不提到日本的索尼公司，该公司在拟人机器人方面做出的贡献是有目共睹的，从最初的 AIBO 到 SDR-4X、SDR-4X II 再到 QRIO，无不表现出非凡的创意。

索尼公司的工业机器人产业始于 1980 年，1997 年启动拟人步行机器人研究计划，样机取名 SDR（Sony Dream Robot，索尼梦幻机器人），以梦幻般的演技为目标。起步虽晚，但索尼公司能利用后发优势，注意吸取前人的经验，特别是早稻田大学的成果。索尼 AIBO 智能玩具狗享誉世界，是四足智能机器人的杰作。索尼拟人双足步行机器人的构思与 AIBO 一脉相承，即供娱乐表演的商品，也显示出后来居上的势头。

2002 年推出的 SDR-4X 新型机器人强化了人机对话功能，提高通信能力，并且在运动性能方面做了改进，以缓和摔倒时的冲击力，如图 1-8 所示。

SDR-4X 具有以下特点：

(1) 尺度小巧、体态轻盈 出于玩具化、商品化的初衷，SDR-4X（见图 1-8）的外形尺寸（高×宽×厚）仅为 50cm×22cm×14cm，这带来诸多优点，如安全、节省能源、便于携带等，然而也伴有若干问题，如倾倒过程加快，控制响应时间更短等。索尼公司创新研究所所长土井利忠认为，下一代产品的目标是身高为 30~40cm，自重为 2~3kg。

(2) 全身协调运动控制 与 ASIMO 相反，SDR-4X 主要靠上身表演动作，下身仅仅服从跟随。由于在腰关节特设了两个自由度，因此她能表演展臂、劈叉、踢球、倾倒后自站立、优美舞姿等。

(3) 伺服电动机的机械-电气-驱动-控制的一体化 SDR-4X 安装 ISA 系列的 3 种体积小、重量轻、精度高、内藏减速器和专用集成电路（ASIC）驱动控制电路板的专用 AC 伺服电动机，真正实现了电动机集成技术。该系列自制电动机的重量仅是同类传统电动机的 $1/4 \sim 1/5$ 。

(4) 智能化 SDR-4X 与 AIBO 机器狗采用相同的 OPEN-R 系统软件。不同的是需要进一步开发借助身段、手势、声音等在一定程度上与人交流的功能，以接近人类的丰富表达力。其中视觉和听觉不可或缺，而语音能力尤为期待和困难。

2003 年 3 月 24 日索尼公司发布了新一代机器人模型——SDR-4X II，如图 1-9 所示，它是 SDR-4X 的升级产品，可以在失去平衡时自动校正，或者对旋转方向或者对前后左右作调整。跌倒时，它还可以自己站起来，重新恢复正常状态。另外，SDR-4X II 还可以识别位置，通过内存的地图自行到达目的地。对于颜色它也能很好地识别，



图 1-8 SDR-4X 双足娱乐机器人



图 1-9 双足行走机器人 SDR-4X II

能够根据颜色判断地点。在语言方面，SDR-4X II 增加了语音辨别、合成用 CPU，可以实现更精准的语言连续识别。在记忆上，它能记住更多的对话。

2003 年 12 月 11 日在日本东京“索尼科学探索”的媒体预展会上，索尼公司又推出了索尼 QRIO——世界首台会跑的双足步行机器人，如图 1-10 所示。索尼定义的“跑”的概念是指机器人行走时双足处于离开地面的非接触状态（Loss of Contact），并不是那种一定要某只脚接触地面像竞走那样的“快步走”。据索尼公司介绍，此前世界上没有可以在不接触地面的状态下行走的机器人。此次开发的双足行走机器人在行走时可以约有 20ms 的不接触地面的时间。该机器人不仅可以行走，而且可以跳跃，在跳跃状态下不接触地面的时间可达 40ms。行走速度为 14m/min。如果假设其大小与人相同，换算过来相当于时速 0.84km/h，给人以缓缓慢跑的印象。

索尼公司的 QRIO 是世界上首个会跑的人形机器人。为防止机体的关节部分挂住用户的衣服、手指等，还特意在关节部位采用了圆弧设计。QRIO 有着良好四肢协调性能。之后，索尼公司在这个基础上强化了人机对话功能、提高通信能力，进一步在运动性能方面做了改进，以缓和摔倒时的冲击力，于是“QRIO”诞生了。这个机器人可以漫步、跳舞，并可做很多高难度动作，它甚至可以指挥一个小型乐队。

“QRIO”的运动性能与通信能力方面都增加了新功能。如强化了双足行走功能、跌倒后自己站起来的功能、识别用户的功能等。追加的这些功能均是针对普通用户设计的。为实现上述功能，QRIO 上内置的两个微处理器分别作为运动中枢和思维中枢，分管运动功能与思维功能。这些性能的设计，很大程度上以在普通家庭中的共同生活为目标，即便行走路面凹凸不平，或者有台阶或斜坡，也能照常行走。

此外，索尼公司都将机器人与音频相结合的表演作为机器人展示中的重要部分，如 QRIO 的太极拳表演，用节奏不强的音乐和动作配合太极拳、千手观音等节奏性不强的动作，以此来弱化节奏在音乐表演中的作用，这种表演方式尤其适合中国传统民乐和古曲的音乐的动作表演，如图 1-11 和图 1-12 所示。



图 1-10 跑步中的索尼新型 QRIO



图 1-11 QRIO 随音乐进行的太极拳表演

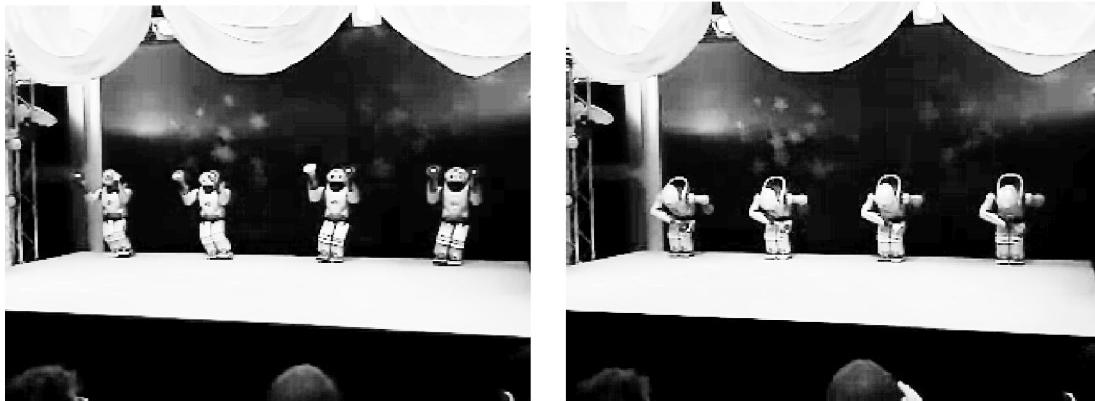


图 1-12 QRIO 随音乐进行的铃铛表演

1.2.1.3 其他

近藤 (Kondo) 公司研发了专门用于拟人机器人的伺服电动机，并将其用于开发低价拟人机器人。他们将机器人结构彻底简化，并将自由度总数降至最少（17个），开发了当时最轻、最小的 KTR21 仿人机器人，该机器人市场价开创了最低价的记录。KTR21 虽只有 17 个自由度，但可以完成走路、体操（俯卧撑、前滚翻、后滚翻、侧手翻、头手倒立）等动作，它可用角度输入法或示教方法编程，可用于教学和科研，如图 1-13 所示。在 2004 年 12 月近藤公司继续努力下，又开发了 21 个自由度的 YDH2EZA 拟人机器人，标价仅为 92400 日元，性价比更好。

目前，商品化的仿人机器人还只有日本在生产。在日本已经形成多家公司竞相研制拟人机器的气候，所研发的仿人机器人也呈现多种型号、各种级别并存的局面。

日本 Speecys 公司推出了能够利用装置分析软件进行动作模拟的“Real Motion System”双足行走机器人开发环境。可使用三维 CAD 软件“SolidWorks”（美国 SolidWorks 公司出品）及作为其插件的 COSMOSMotion 运动分析软件，对该公司“Speecys (SPC-003)”新型双足行走机器人的动作进行模拟。主要面向大学等研究领域销售。

Real Motion System 包括 SPC-003 的三维模型，为该模型设置动作后进行分析。由于能够考虑重心、扭矩和惯性等因素，因此能检测在所定义的动作过程中是否会摔倒，零部件之间是否有干扰。

过去，都是直接将动作编辑器定义的条件直接输给机器人，使之运动。但是，动作编辑器只是设定各关节的转动角度及其次序，并不能考虑平衡等因素。所以很难定义出可使机器人保持平衡、不会摔倒的动作，在定义动作时都需要反复摸索。据称，在定义动作的过程中有时还会损坏机器人。如果使用 Real Motion System，由于能够全部在桌面上进行研究，因此就能防止上述问题。另外，修改部件形状，或追加新的部件时，也能预先利用 SolidWorks 和 COSMOSMotion 验证平衡性，因此能够提高开发速度。

1.2.1.4 日本机器人研究主要代表人物

古田贵之，日本科技振兴事业团（ERATO）北野共生系统工程研究员、青山学院特聘研究员。1968 年出生，1996 年任青山学院 E-sys 研究室主要成员，独自一人开发了 Mk 系列步行机器人研究平台，被誉为机器人开发的年青奇才。

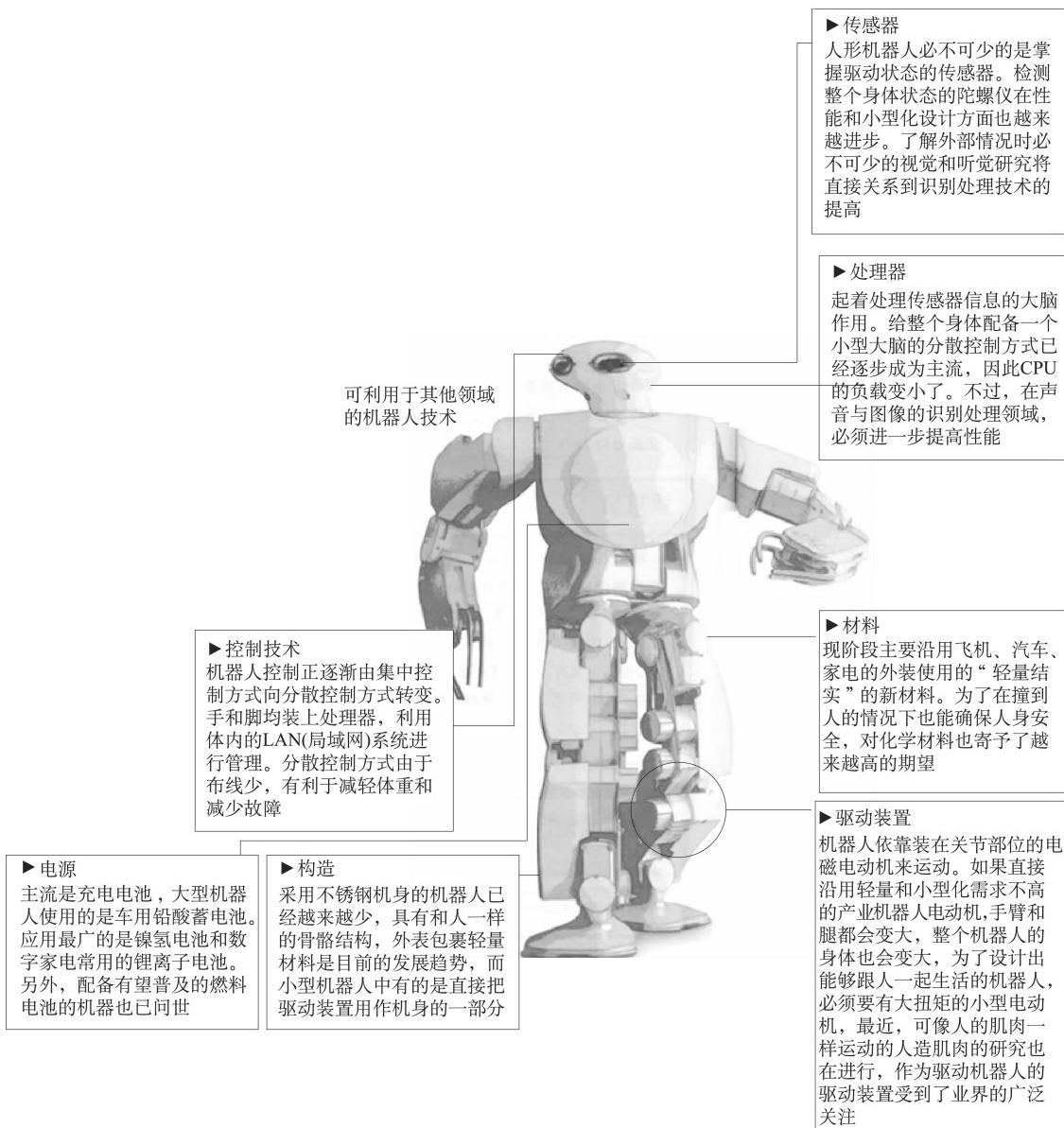


图 1-13 “Speecys (SPC-003)” 新型双足行走机器人

注：参考日本千叶工业大学“morph3”而设计的机器人示意图

广瀬真人，现任 ASIMO 开发室经理、主任工程师。1956 年出生，1980 年硕士毕业，1986 年加入本田技术研究所，是 E (E0 ~ E6) 系列机器人、P (P1 ~ P3) 系列机器人及 ASIMO 机器人的研发者。

石田健藏，索尼创新研究所第 2 部长。1974 年早稻田硕士毕业，1976 ~ 1977 年斯坦福大学客座研究员，此后至 1996 年从事工业机器人和高速插件机的研发应用，1997 年开始负责 SDR-4X 机器人的研制。

北野宏明，现任 ERATO 的北野共生系统工程负责人，1961 年出生，工学博士。该工程

包括“拟人机器人与人工智能”和“分子生物学的生命系统”两大项目，互有关联。1993年获得被誉为人工智能诺贝尔奖的“The Computers and Thought Award（计算机与思维奖）”奖，1999年当选新闻周刊《News week》21世纪100位杰出人才。他是机器人足球世界杯主席。

高西淳夫，1980年毕业于早稻田大学，1985年获硕士学位，1988年获博士学位，学生时代即参与WL-9DR的研究，提出“准动态步行步态”的概念。一直从事类人双足步行机器人的研究^[5]。

1.2.1.5 韩国的机器人发展

与此同时，韩国正在急起直追。2004年12月27日，韩国先进科技研究所（KAIST）发表了HUBO^[10]机器人。这个机器人高为125cm，重为55kg，具有41个自由度，10个手指都能独立活动，双眼可以转动。研究经费耗资2000万韩元，就在两天后，韩国另一家研究机构韩国科技研究所（KIST）也发表了NBH21机器人，这个机器人高为150cm，重为67kg，据称机器人的软件可通过网络从服务器得到更新，如图1-14所示。

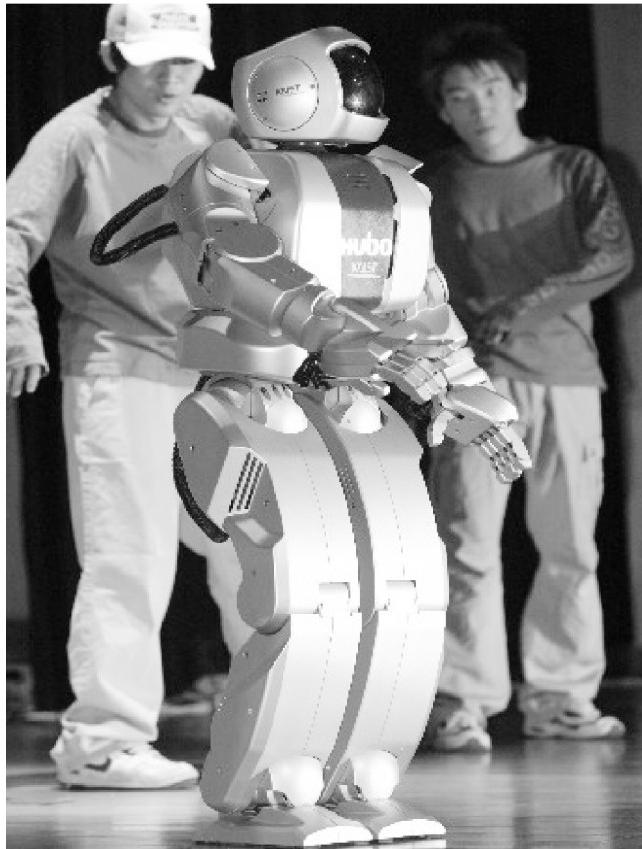


图1-14 韩国的HUBO拟人机器人

1.2.2 美国和欧洲

(1) 麻省理工学院机器人 1999年美国麻省理工学院研制出了COG拟人机器人，由头、躯干、胳膊及双手组成。它是人和人工智能等领域的一个平台。2005年麻省理工学院研制出具有29个自由度的Domo机器人。同年该学院还研发出双足学习机器人，采用“被

动动力学 (Passive Dynamic)" 理论设计, 部分关节没有电动机和控制器, 却能实现非常类人的运动。

麻省理工大学研制的 Kismet 机器人, 如图 1-15 所示。

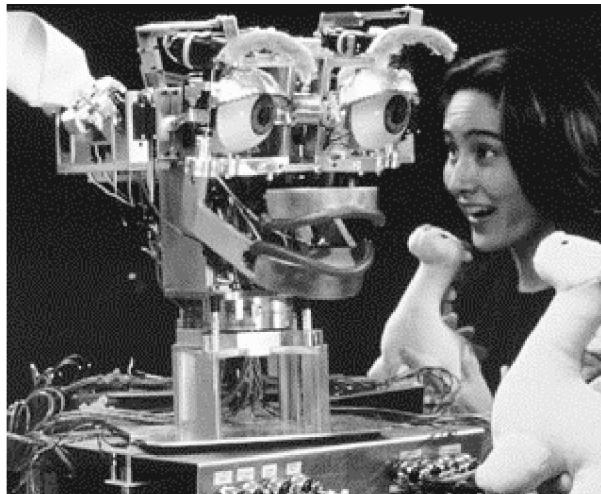


图 1-15 Kismet 机器人

Kismet 总共有 21 个自由度, 如图 1-16 所示。3 个自由度控制眼球注视的方向, 3 个控制头部的方向, 其余 15 个控制面部表情 (眼睑、眉、嘴唇和耳朵)。Kismet 的眼部有 4 个 CCD 照相机、2 个传声器。

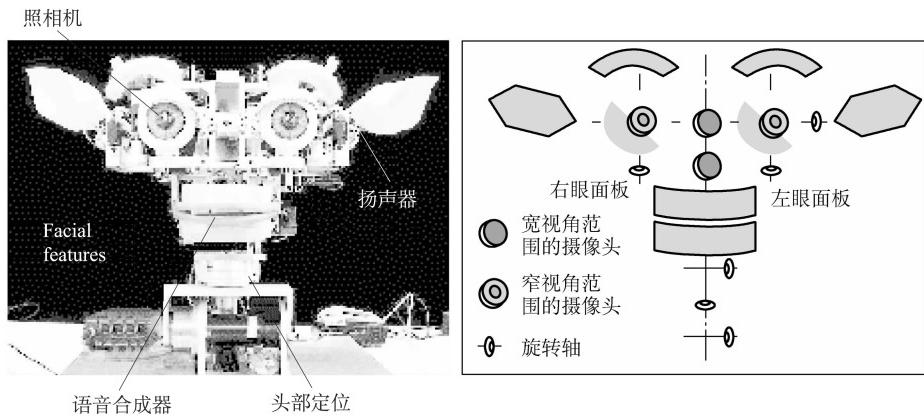


图 1-16 Kismet 机器人组成

其功能有: 视觉跟踪, 采用图像处理技术, 跟踪物体; 面部可以呈现一些表情, 主要的面部表情运动有眉毛和嘴; 可以讲话, 通过语音识别和人工智能技术, 实现与人进行简单的语音交互功能; 表达情感, 这个功能是 Kismet 一个非常突出的功能, 根据人的表情、环境等表达自己的情绪, 例如高兴、惊讶、沮丧等。图 1-17 所示是 Kismet 机器人情感表达空间, 表示了 Kismet 机器人的情绪分布情况。根据机器人的情感可以表达出来机器人情绪状态, 意味着机器人将不再是一个冰冷的金属装置, 它们也慢慢地具有了类似人的情绪, 表达类似人的情感。这是机器人研究的一个重大进步。

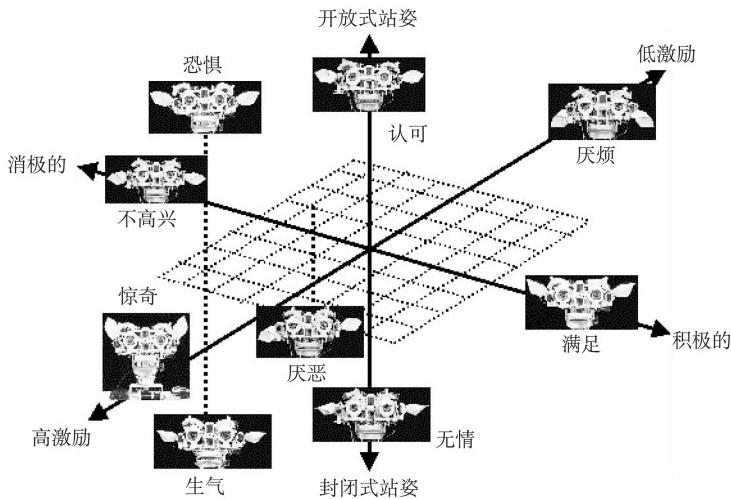


图 1-17 Kismet 机器人的表情表达空间

Kismet 不光具有上述一些功能，还具有学习能力，通过人的不断训练，它可以通过学习不断地完善自己的表达，可以看出来，Kismet 机器人是一个聪明的家伙，虽然他没有强健的身体，但是它的大脑开始不断地强大起来。

(2) Rabbit 机器人 法国格勒诺布尔自动化实验室的 Rabbit 是世界上第四个可以跑动的双足步行机器人，Rabbit 拥有较简单的机械结构，它没有脚，只有一对“脚踝”，即一对枢轴，这样的设计使得 Rabbit 在向前迈步时，整条腿能够绕着枢轴转动，如图 1-18 所示。其原型高 1.425m，重 36kg。Rabbit 是按照机械结构最简单来运动的。

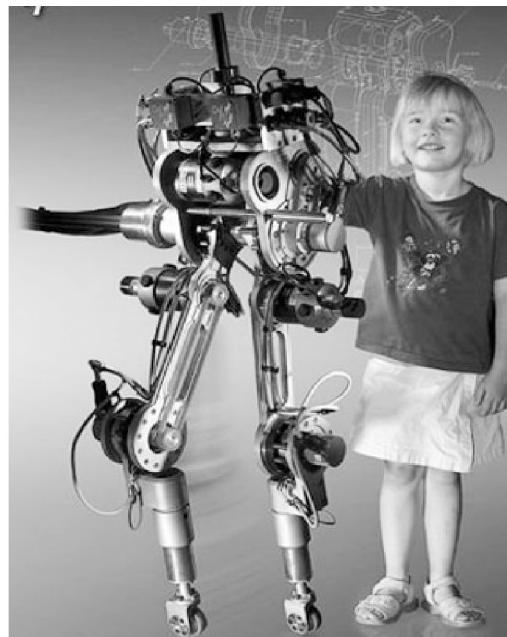


图 1-18 法国格勒诺布尔自动化实验室的 Rabbit 机器人

此外，2004 年 NASA 科学家 Dr. Mark W. Tilden 依照（生物学、电子学、美学、机械学）（Biology, Electronics, Aesthetics and Mechanics, BEAM）四大原理与技术设计开发出 Robo Sapi En 双足娱乐型机器人。

德国慕尼黑技术大学的 Johnnie，开发了具有 17 个自由度，身高为 1.8m，重量为 40kg（不带电源），其中上肢有 2 个自由度，在行走时摆动来补偿重力矩变化。其关节由直流电动机加齿轮传动机构来驱动，关节转角和速度由增量码盘测量。采用了数字传感器以最小化数字扰动，足部装有六维力传感器。主机采用 P4 2.8GHz 处理器，利用 PCI 总线实现与传感器和驱动器间的数据传输。整个控制算法运行于 RT-Linux 实时操作系统。其行走时速为 2.2km/h，它是 2003 年当时世界上行走速度最快的双足步行机器人，也是世界上最高的双足步行机器人，能实现平稳行走，进行图像识别和自主避障功能。

2005 年 2 月 21 日召开的第 171 届美国科学促进会的年会上，由美国康奈尔大学、麻省理工学院和荷兰代尔夫特理工大学的科学家分别组成的研究小组，展示了他们研制出的能以人类步态直立行走的被动动力行走机器人，被认为代表了当前被动动力机器人研究的最新成果，如图 1-19 所示。

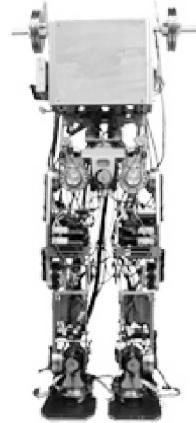


图 1-19 WABLAN-4 机器人

1.3 国内研究状况

与国外相比，我国从 20 世纪 80 年代中期才开始研究拟人机器人。在“七五”期间，国防科技大学和哈尔滨工业大学等单位分别进行了研究。在“863”项目支持下，国内的机器人研究有了长足的发展。

(1) 北京科技大学机器人 北京科技大学在研究机器人方面取得了巨大的成绩，2005 年北京科技大学研制了双足步行机器人，如图 1-20 所示。它是具有 17 个自由度的，并可在平面上做各种拟人动作的双足步行机器人，采用了智能技术，通过无线传输，可使多个机器人同时进行表演，可以模拟人类的前进、后退、转弯、下蹲、侧翻、打太极拳、体操表演、直立前行行走、前滚翻、后滚翻、做俯卧撑、伏地起身、上下楼梯。

2006 年，北京科技大学独立研制了具有人类外观特征的大型机器人乐乐（见图 1-21），他具有人脸识别、语音识别、动作表演、生活常识、背诗、聊天、旅游常识、唱歌、生活常识、脑筋急转弯等多种功能。随后北京科技大学又研制出了大型机器人娜娜（见图 1-22），并参加了天津科技馆的展

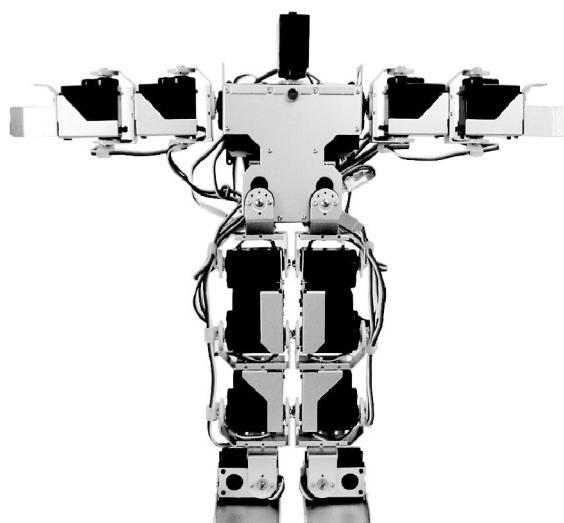


图 1-20 小型双足机器人

出，娜娜高度和人的身高相仿，能模仿人的各种基本动作，同时具备语音识别与语音合成功能，可实现预定内容的人机交互；主机器人可以和客人进行日常用语对话，进行场馆的内容介绍，可以实现不同声调的变换，如男声、女声、卡通声音等。

北京科技大学于2007年初在重庆展出了又一拟人机器人爱因斯坦（见图1-23），具有语音对话能力，男声，具备数据库更新能力，可以随时更新内容；能模仿人的各种基本动作，如摆头、点头、嘴的张合、眨眼、两臂前摆、外摆、大臂前摆、小臂伸曲、转腕、摆手等动作，还可与展区的其他机器人配合进行互动表演。

2007年12月，北京科技大学在大庆科技馆展示了由大型机器人科大一号（见图1-24）和双足步行机器人（见图1-20）组成的机器人团体表演，科大一号可以通过传感器网络感知环境，有方向感，可实现一定范围内的机器人全方位自主运动，机器人的作业臂可以灵活地运动。除此之外，它还可以进行跳舞、唱歌、背唐诗等表演，脸部采用动画显示表情，可以作出多种表情；通过多媒体计算机还可进行查询及场景解说。双足步行机器人主要是通过控制舵机来模仿人的身体结构和关节运动，可以在平面上做各种拟人动作，采用了智能技术，通过无线传输，可以多个小型机器人同时进行表演，可以实现人的行走、做俯卧撑、鞠躬、踢球等多种动作的模仿。



图1-21 乐乐

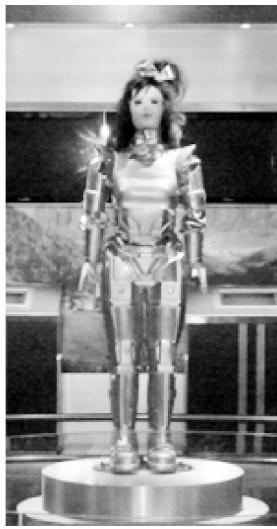


图1-22 娜娜



图1-23 爱因斯坦



图1-24 科大一号

(2) 哈尔滨工业大学机器人 哈尔滨工业大学是较早研制双足步行机器人的高校，它开始于1985年，早期的机器人没有头部和双臂，到1995年研制成功了HITⅠ、HITⅡ和HITⅢ^[10]三个型号。1995年研制成功的HIT机器人的双腿具有12个自由度，踝关节由两个正交电动机构成，同时实现两个自由度。2004年6月，哈尔滨工业大学研制成功能用脚踢球的双足拟人足球机器人。

(3) 国防科技大学机器人^[11] 2000年，在国防科技大学诞生了我国独立研制的第一台具有人类外观特征、可以模拟人类行走与基本操作功能的“先行者”平面拟人机器人。“先行者”高为114m、重为20kg，具有人一样的身躯、脖子、头部、眼睛、双臂与双足，并具备了一定的语言功能。目前国防科技大学研制的新型拟人机器人有36个自由度，其中每条

腿各 6 个，每个胳膊各 6 个，手部 5 个，头部 2 个，下肢各个关节有位置传感器，足部有多维力矩传感器；整个控制系统、电源集成在机器人本体上。采用一种以 DSP 为核心、CAN 总线为通信标准的新型控制结构进行控制。

(4) 北京理工大学机器人 我国第一个真正意义上的 BHR-1 拟人机器人在 2002 年 12 月通过了国家 863 项目组的验收。该机器人按照亚洲人的标准，身高为 1158mm，体重为 76kg。它全身有 32 个关节，使其活动起来比较自如，步幅为 0.133m，速度为 1km/h。机器人手、脚可以完成 360° 的旋转，比人更灵活。除了行走，它还能蹲下、站起、原地踏步、打太极拳。这个机器人还会腾空行走，并能根据自身的平衡状态和地面高度变化，实现未知路面的稳定行走。在系统集成、步态规划和控制系统等方面实现了重大的突破。目前 BHR-1 已经研制到第 3 代“汇童”，高度为 116cm，重量为 63kg，突破了仿人机器人的复杂动作设计，在国际上首次实现了模仿太极拳、刀术等人类复杂动作。2005 年 9 月 18 日，由北京理工大学牵头、中科院沈阳自动化所、兵器工业集团惠丰机械有限公司、航天工业集团红林机械厂、中科院北京自动化所等单位参加，通过 3 年的奋斗，研制的“汇童”拟人机器人，如图 1-25 所示。该拟人机器人是科技部“十五”863 计划先进制造与自动化领域机器人技术主题的重点项目，同时得到了国防科工委国防预研专项、国防基础科研重点专项、北京理工大学 985 工程科技创新专项等大力扶持。“汇童”身高为 1.60m，体重为 63kg，全身共有 32 个自由度，配备力传感器、视觉传感器、加速度计、倾角计、编码器等各类传感器共 44 个，可模仿人类前进、后退、侧行、转弯、上下台阶、太极拳、刀术等各种复杂动作。“汇童”是具有视觉、语音对话、力觉、平衡觉等功能的自主知识产权的拟人机器人，功能达



图 1-25 “汇童”双足机器人

到了国际先进水平，它的成功研制标志着我国成为少数掌握集机构、控制、传感器、电源于一体的高度集成技术的国家。

(5) 清华大学机器人 清华大学设计的THBIP机器人，本体结构类人，共设置32个自由度，头部2个，上肢12个，下肢12个，手部6个。躯干部装有陀螺姿态传感器，脚底和手腕装有六维力传感器；各关节装有位置传感器。机器人采用模糊非线性控制，成功实现了无缆连续稳定的平地前进、后退、侧行、连续上下台阶行走，以及端水、太极拳和点头等动作。

除以上单位外，上海交通大学等单位也在进行相关研究^[6]。

1.4 其他相关技术

拟人机器人研究在很多方面已经取得了突破，如关键机械单元、基本行走能力、整体运动、动态视觉等，但是离我们理想中的要求还相去甚远，还需要在拟人机器人思维和学习能力、与环境的交互、躯体结构和四肢运动、体系结构等方面进行更进一步的研究。研制与人类外观特征类似，具有人类智能、灵活性，并能够与人交流，不断适应环境的拟人机器人一直是人类的梦想之一。

1. 躯体结构和四肢运动

毫无疑问，拟人机器人行动的多样性、通用性和必要的柔性是“智能”实现的首要因素。它是保证拟人机器人可塑性和与人交流的前提。拟人机器人的结构则决定了它能不能为人所接受，而且也是它像不像人的关键。拟人机器人必须拥有类似人类上肢的两条机械臂。这样不仅可以满足一般的机器人操作需求，而且可以实现双臂协调控制和手指控制以实现更为复杂的操作，如图1-26所示。拟人机器人要具有完成复杂任务所需要的感知活动，还要在已经完成过的任务重复出现时要像条件反射一样自然地做出反应。

2. 与环境的交互

拟人机器人与环境相互影响的能力依赖于其富于表现力的交流能力，如肢体语言、思维和意识的交互。目前，机器人与人的交流仅限于固定的几个词句和简单的行为方式，其主要原因是：

- 1) 大多数拟人机器人的信息输入传感器是单模型的。
- 2) 部分应用多模型传感器的系统没有采用对话的交流方式。
- 3) 对输入信息的采集仅限于固定的位置，比如图像信息，照相机往往没有多维视角，信息的深度和广度都难以保证，准确性下降。

3. 体系结构

拟人机器人的体系结构是定义机器人系统各组成部分之间相互关系和功能分配，确定单

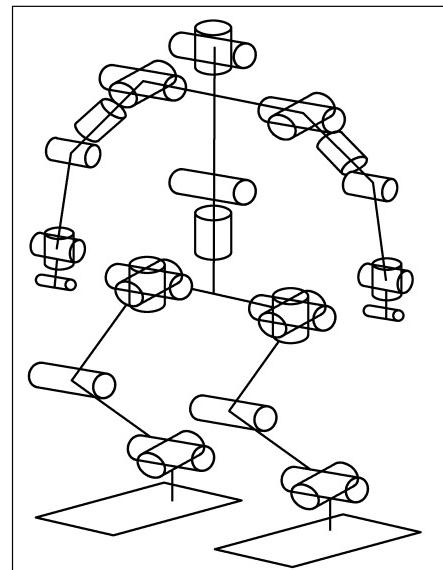


图1-26 拟人机器人的躯体结构

台机器人或多台机器人系统的信息流通关系和逻辑的计算结构，也就是确定拟人机器人信息处理和控制系统的总体结构。如果说机器人的自治能力是拟人机器人的设计目标，那么体系结构的设计就是实现这一目标的手段，如图 1-27 所示。现在拟人机器人的研究系统追求的是采用某种思想和技术，从而实现某种功能或达到某种水平。所以其体系结构各有不同，往往就事论事。解决体系结构中的各种问题，并提出具有一定普遍指导意义的结构思想无疑具有重要的理论和实际价值，这是摆在我们面前的一项长期而艰巨的任务。

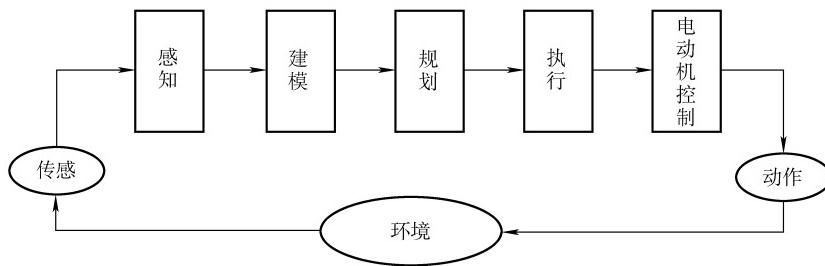


图 1-27 拟人机器人的串行功能分解体系结构

4. 思维和学习能力

现有拟人机器人系统的主要缺陷是对环境的适应性和学习能力的不足。机器人智能来源于与外界环境的相互作用，同时也反映在对作业的独立完成度上。机器人学习控制技术是实现类人机器人在结构和非结构环境下实现智能化控制的一项重要技术。但是由于受到传感器噪声、随机运动、在线学习方式以及训练时间的限制，学习控制的实时性还不能令人满意，仍需要研究和开发新的学习算法、学习方式，以不断完善学习控制理论和相应的评价理论。目前针对机器人学习控制的研究，大都停留在试验室仿真的水平上。

拟人机器人具有人的外观，可以适应人类的生活和工作环境，帮助人类完成各种作业，并可以在很多方面扩展人类的能力，在服务、医疗、教育、娱乐等多个领域都将得到广泛的应用。在人类的历史中，曾经因为我们制造机器的局限性，使得我们不得不去适应机器，而现在我们要让机器来适应我们，这就是拟人机器人的未来发展趋势。

1.5 相关参考资料

1.5.1 学术会议和期刊

专门的拟人机器人学术会议和期刊作为研究工作总结的学术论文除见于一般的机器人会议和期刊外，自 2000 年起由 IEEE/RSJ 召开的《International Conference on Humanoid Robots》和自 2004 年起出版的《International Journal of Humanoid Robotics》季刊提供了专门的园地。

1.5.2 网站

- 1) 韩国拟人机器人网站：Android world [EB/OL]. <http://www.androidworld.com/index.html>

- 2) Honda. Asimo : the Honda humanoid robot Asimo [EB/OL]. <http://world.honda.com/ASIMO/>
- 3) QRIO [EB/OL]. http://www.sony.net/SonyInfo/QRIO/top_nf.html
- 4) Fujitsu. Services & products, humanoid robot HOAP22 [EB/OL]. <http://www.automation.fujitsu.com/en/products/products09.html>
- 5) Kondo [EB/OL]. <http://www.kondo2robot.com/index.html>
- 6) Speecys-products for futuristic entertainment [EB/OL] <http://www.speecys.com/products.html#robotkit>
- 7) Robogarage, KyotoUniversity, Chroino [EB/OL] <http://www.eonet.ne.jp/~robogarage/english/robo/chroino.html>
- 8) Robot labs [EB/OL]. <http://www.robot2labs.jp/top.html>
- 9) MCLab [EB/OL]. http://ohzlab.kaist.ac.kr/robot/KHR23_intro.html
- 10) 北京科技大学机器人网站: www.aprobot.com

1.6 小结

本章详细介绍了机器人的发展史，从第一台可控制的机械手到现在的双足机器人经历了漫长时期，但机器人越来越朝着智能化和拟人化的方向发展。还介绍了机器人的定义、机器人的分类、机器人三定律、机器人技术及发展方向、机器人系统理论，同时还对日本、韩国等外国及国内对机器人的研究概况做了详尽的介绍，提出了研究机器人的科学意义。

参 考 文 献

- [1] 杨敏. 拟人机器人的步态规划仿真研究 [D]. 南京: 南京工业大学, 2006.
- [2] 祁乐, 闫继宏, 朱延河, 等. 小型双足步行机器人的研制 [J]. 机械工程师, 2006 (11).
- [3] 李允明. 国外仿人机器人发展概况 [J]. 机器人, 2005, 27 (6).
- [4] 阮晓钢, 仇忠臣, 关佳亮. 双足行走机器人发展现状及展望 [J]. 机械工程师, 2007 (2).
- [5] 宗光华, 唐伯雁. 日本拟人型两足步行机器人研发状况及我见 [J]. 机器人, 2002 (11).
- [6] 满翠华, 范迅, 张华, 等. 类人机器人研究现状和展望 [J]. 农业机械学报, 2006, 37 (9).
- [7] 国家863计划智能机器人专家组. 机器人博览 [M]. 北京: 中国科学技术出版社, 2001.
- [8] Hirai K, Hiroseand M, Takenaka T. Current and Future Perspective of Honda Humanoid Robot [C]. EEE/RSJint. Intelligent Robots and Systems, 1998, Tokyo: 500-508.
- [9] Hirai K, Hirose M, Takenaka T. The Development of Honda Humanoid Robot [C]//IEEE Int. Conf. on Roboticsand Automation, Leuven: 1998, 1321-1322.
- [10] 杨新海. 双足步行方法与控制研究 [D]. 哈尔滨: 哈尔滨工程大学, 2006.
- [11] 吴立勤. 两足步行机器人步行步态研究及仿真 [D], 上海: 上海大学, 2002.

第2章 双足步行机器人相关理论与方法

2.1 概述

双足步行机器人理论与技术是一个多学科集成的研究领域，涉及智能控制（其中主要包括位置与姿态的稳定性控制、鲁棒控制）、行为步态的自动生成方法、和谐人机交互理论、人工智能与人工情感理论方法等多个方面。下面介绍几个主要研究内容。

(1) 稳定性与控制 稳定性与控制策略是双足步行机器人技术的关键，也是各国学者研究的焦点。具体主要有双足运动的动态稳定与控制机理、双足步行运动的固有鲁棒性机理、实时步态规划与控制等几个方面。只有解决了这些问题，双足步行机器人才能实现自然、顺畅地行走，从而更好地完成任务。2006年10月在中国苏州举行的机器人大赛上，所展出的双足步行机器人已经具有较好的稳定性，不但具有顺畅的行走能力，而且在不慎摔倒时还能够独立地站立起来继续行走。

(2) 开发新型关节驱动器 双足步行机器人的自由度是其完成动作质量的保证，自由度越多，其动作越协调、顺畅。目前研制的双足步行机器人自由度最少的也有十几个，最多的达到几十个。自由度增多的同时，对驱动器的要求也越来越高，已不局限于传统的几种方式，形状记忆合金驱动、压电陶瓷驱动等方式逐渐应用到机器人领域，这些驱动方式以驱动速度快、负载能力强等特点已经逐渐取代了传统的驱动方式。

(3) 改进人机接口设计 机器人要与人类共存并合作，做人类做不到的事，开拓机动性的新领域，从而对人类社会产生附加价值，而并不是完全取代人。目前，双足步行机器人还不能摆脱人的控制而独立工作，因此良好的人机协调系统在机器人的工作中将起到非常重要的作用。

(4) 双足步行机器人行为发生器 一个机器人行为的展示，主要取决于设计者对机器人行为的规范和控制方法。实现机器人控制的方法有很多种，例如火星漫步者号利用的是“行为基础控制”方法，通过了两种方式来实现“行为基础控制”：一个是“模糊逻辑”，另一个是“神经网络”。这两部分的主要区别为，应用“模糊逻辑”可以通过现有的、不断丰富更新的知识来完成任务，而应用“神经网络”可以从零起步，随着时间的推移逐步实现知识的学习和累积。

机器人的行为是一个非常复杂的概念，它建立于人工智能、人工心理等学科基础之上。而现在所研究的机器人还不具有“思考”的功能，也就是说，这个双足步行机器人所完成的所有动作都是根据我们提供的动作数据进行的。那么如何产生这些动作的数据呢？这就是所谓行为发生器需要解决的问题。

2.2 双足步行机器人相关理论与方法的形成及发展

从第一台双足步行机器人诞生到现在，双足步行机器人的理论及方法经历了一个长期缓慢的发展过程。

2.2.1 稳定性与控制方法

1. 传统有动力驱动机器人

双足步行机器人的研究始于 20 世纪 60 年代，在 1968 年，美国通用公司的 R. Smosher 试制了一台名为“Rig”的操纵型双足步行机器人，揭开了双足步行机器人研究的序幕。但该机器人只有踝和髋两个关节，操纵者靠力反馈感觉来保持机器人的平衡。

1968 ~ 1969 年，南斯拉夫的著名科学家 M. Vukobratovic 提出了一种重要的研究双足步行机器人的理论，即零力矩点（ZMP）稳定判据，并研制出世界上第一台真正意义的双足步行机器人。随后各国的学者和研究机构纷纷展开双足步行机器人的研究工作。

日本早稻田大学的双足机器人研究始于 1968 年，该大学的加藤一郎教授所领导的课题组相继研制了 WL 系列、WABIAN 系列双足机器人。最新的研究成果为 WL-16 和 WABIAN-2 机器人，WL-16 机器人腿部采用一对 6 自由度的并联机构。由于并联机构刚度高，许多金属部件用聚乙烯树脂代替，齿轮、电动机等重量都得以减轻，可以在不平的地面上稳定行走。本田的 ASIMO 机器人通过事先预测到下肢转弯后重心向外侧倾斜多少等重心变化，可以使得从直行改为转弯时的步行动作变得连续流畅。而索尼的 QRIO 机器人在设计的过程中，将此前的用于步行的算法中增加了可即时控制机器人的跳跃方向和在空中时可保持平衡状态的姿势控制等的算法。

但是，传统有动力驱动机器人不能够动力平衡，它的行走是预先编好程序的，如果你让它在同样一个地方走两次，它在第二次会将它的脚放在与第一次同样位置上。当然前提是地面要坚固并且平整。而且，动力驱动双足行走机器人能量消耗大，这个问题一直是研究者考虑的重点。

2. 被动动力行走机器人

在任何一个特定时刻的行走过程中，我们人类不得不考虑（或者至少你的身体这样做了）落脚的方向，将脚放在正确的位置，使你能够沿你想的方向运动。对我们而言，实现这个过程很容易，但是求解这个问题很困难。当机器人脚落下时，意味着它的重心并不在脚（或两脚）的中间，偏差增长很快。如果机器人的不平衡达到几厘米时，就不太可能恢复了，因为也不能通过快速移动肢体进行补偿。像人一样高的东西会非常迅速、无法挽救地失去平衡。

美国航空工程师 Tad McGeer 在 1989 年提出一个新的理论，完全不用驱动也不用控制的机器人也能够实现稳定的步行运动，进而提出了“被动动力步行”概念。动力平衡方式行走，落脚实际上是经过控制的，使得机器人的行走看起来相当平稳。此后许多学者对被动动力行走机器人进行了深入的研究，并且研制出被动动力机器人样机。2005 年 2 月 21 日召开的第 171 届美国科学促进会的年会上，由美国康奈尔大学、麻省理工学院和荷兰代尔夫特理工大学的科学家分别组成的研究小组，展示了他们研制出的能以人类步态直立行走的被动动

力行走机器人，被认为代表了当前被动动力机器人研究的最新成果。

被动动力行走机器人内置多个传感器，使其能在20min内学会像人一样走路。传感器能以200次/s的速度测量机器人每个动作的倾斜度和速度，然后向其他电动机下达新指令来调节身体姿势。传感器还指示制动器控制机器人脚踝上弹簧的压力，使其整体平稳向前。每走一步，制动参数都会有所改变，所以机器人可以在任何路面上走路，并调节走路的姿势。

被动动力步行机器人的最大优点在于，能量消耗低却能够实现像人一样有自然的步态。已经得到了国外众多学者的深入研究，并取得了巨大的成果。但当前对被动动力步行机器人的研究还不成熟，如稳定性、倒行等还需要进一步的深入研究。

2.2.2 双足步行机器人行为发生方法

人的行为动作多种多样，而机器人的行为是一个非常复杂的概念，它建立于人工智能、人工心理等学科基础之上。目前双足步行机器人所能做的动作都是人们根据人的各种动作编写并调试出来的，所以现在双足步行机器人还不具有“思考”的功能，也就是说，根据我们提供的动作数据，这个双足步行机器人才能完成所有动作。那么如何产生这些动作的数据呢？这就是所谓行为发生器需要解决的问题。

所谓行为数据库就是一个存储了机器人基本动作单元的数据库。从一个特定的动作分析，一个动作是由若干个关键帧动作组成的，可以将一些关键帧作为动作单元保存起来，运行时，可根据不同的动作选择出不同的关键帧，拼凑出一个完整的动作。把一些简单的动作通过不同的组合可以得到一些复杂而有意义的动作。

机器人的行为是一个非常宽泛的概念，它受到多个学科的影响，特别是人工智能学科的发展对机器人的行为有着非常重要的作用和意义。

2.2.3 人机接口方法

自20世纪80年代开始崛起的各种应用背景下服务于人类的机器人理论研究及应用技术(RT)，是人类在制造业领域、信息技术领域、认知心理学领域等多学科共同进步的一个重要里程碑。融合多学科领域的人与机器人交互及共存机制和技术是服务机器人研究内容的重要组成部分，是在时间、空间坐标系上感知、决策、规划、协同、行为等多种理论研究和分析方法的有机融合。一般来说，人与机器人和谐共存的研究主要集中在交互过程中多种模式状态分析和理解的方法及实现等方面，如生物特征信息提取、多模态信息融合、机电系统协同控制理论与方法，以及软硬件协同设计技术等^[1]。由于交互过程中的内容是多种多样的，每一种类型应用背景下又有各自的特点，对任意类型内容的人与机器人交互技术研究是一件非常困难的事情，因此往往针对特定应用和利用专业领域知识对给定类型的人与机器人交互及共存中的问题进行分析、理解和实现方法等方面的研究^[2]。20世纪90年代初及本世纪初期的近十余年中，研究者们对人与机器（机器人、计算机等）的交互过程中多种状态分析、理解和实现等进行了大量的研究。近几年来，针对在典型背景下的拟人机器人社交能力的理论研究和应用技术逐渐成为研究热点，研究认知与交互、情绪与交互的关系正在成为该领域的前沿问题^[3]。这主要是由于拟人机器人的外貌、表情、语言、肢体行为对人类用户来说至关重要，因此对拟人机器人的需求推动着该方向研究的发展。另一方面，拟人机器人与人

类用户在共存与交互过程中信息的认知特征和情绪特征的不确定性和多样性等因素，吸引着越来越多的研究者投入到这一领域^[4]。

从研究机构方面来看，早在 20 世纪 70 年代日本提出了“感性工学（Kansei Engineering）”的概念，就是将感知、认知科学与相关应用领域结合起来的基础理论与应用技术，是在认知科学的基础上，通过分析人类的感觉、知觉等信息特征，把人类所特有的认知需要加入到商品设计、制造中去。它是一门从工程理论科学的实现角度，能给人类带来喜悦和满足的商品制造的技术科学。在文部省（日本中央政府行政机关，全称“文部科学省”）主导下，20 世纪 90 年代以来，日本各个学术领域积极引入感性工学的观念，尤其在诸如机器人科学与技术、人工智能等学科^[5]。如 20 世纪 90 年代日本的索尼公司，以此为基础研制出 AIBO 狗机器人，拥有先进的性能和品质，并且是第一个实现规模商品化的宠物机器人，为有社会交互能力的机器人及相关的研究打开了实用的空间。其后大量的宠物机器人不断涌现，其共同特点是对外界刺激有反应能力，能同儿童进行生动的交流。这种交流除了语言，还有面部表情、语调和身体姿态等^[6]。而随后 NEC 公司所开发的 PaPeRo 儿童家用机器人具有听觉、视觉，是个有个性、有表情的小型机器人，是一个可以与之一块生活、能记住小孩的喜好、让每个人都能在不知不觉间享受相互交流所带来的好处的“伴侣”，是 NEC 开发的“未来家庭的无键盘计算机”。它具有散步模式和对话模式，在没人与之对话时进入散步模式，可以在房间随意散步。而在看到有小孩时即进入对话模式，可与儿童交流。同时能识别约 650 个单词，能说 3000 句话，能辨认人的脸。可以不需操作键盘，一边与之对话，一边进行许多其他的工作，如上网、收发邮件、接收信息，自动传达必要的信息等；利用它的摄录像功能，在孩子们之间传话；能够与儿童猜谜语、跳舞、叫人起床、遥控电视等^[7]。从中不难看出，正是以“感性工学”的技术和方法为核心，人类用户与机器人之间可以进行快乐和舒适的交流。

日本大阪大学的石黑浩教授研制的 Ac-troid 机器人能用坐姿与人类交流。“她”全身安装了 31 个动作器和 11 个触觉传感器，有拟人的眼球、睫毛、会动的嘴唇和人造肌肉，具备拟人表情，能听懂 4 万多个中文、英文、日文和韩文语句，并配备适合于 2000 多种答案的面部表情，可以容纳更多的元部件，做出更复杂的动作。石黑浩教授应用认知心理学有关理论，将基于行为主义的机制与人工智能联系起来，形成了自己独特的交互技术和方法，并借助奇异地理论作为评价方法，使得研制的 Ac-troid 机器人能用坐姿通过多种语言、丰富的表情、恰当的手势等方式与人类进行自然、友好的交流^[11]。

日本早稻田大学开发的儿童玩伴机器人有视觉、听觉、触觉和嗅觉传感器来感知外界刺激信号。根据所建立的儿童大脑结构化模型，通过外部和内部的刺激相应地改变其情绪状态，而后用多种方式表达出来，并通过非线性计算所产生的内部钟来表述心境矢量的激活成分。

美国斯坦福大学传播研究所的两位教授克利福德·纳斯和拜伦·里弗斯（Clifford Nass & Byron Reeves）针对人与机器人的交流问题进行了一系列研究，通过理论研究与实验分析得出如下结论：人与机器间的联系有自然性和社会性，在人机交互中所需要解决的问题同人和人交流中的是一致的，关键就是赋予机器人“基于人类认知心理的情感能” 的能力^[8]。

美国麻省理工学院（MIT）所研制的 Kismet 机器人，通过所建立的环境、内部刺激和行

为动作的认知心理模型，对外界输入的刺激和内部需要进行综合判断，从而引起表现行为的各种变化。具有与儿童用户相似的行为方式和能力，比如模仿父母与孩子之间表示情绪状态的反馈方式，小孩向父母表达需求和愿望的方式，以及儿童自我学习与人和环境交流的方式等。

德国慕尼黑大学所研制的 EDDIE 交互机器人，这些执行机构根据特定的人脸运动编码系统来设置，每个表情状态通过数学映射算法将计算出的结果作为给定信号来驱动执行机构进行相应的表情合成，并在与儿童的交流测评中，得到了比较满意的结果^[9]。

英国赫特福德大学的研究者将信息科学、认知心理学等多学科结合，在人与机器人交互过程中采用生物启发式的情感建模，并引入了个性的概念，包括感知个性和表达个性。其中，感知个性决定刺激如何影响机器人情感状态，表达个性则影响机器人表情和颈部运动。将此模型算法应用于实际的交互过程之中。根据隐马尔可夫等随机统计模型建立起语言模型和声学模型，并由此获得语言辨识引擎，同时通过基于人类声道的有源滤波模型，实现了语音合成系统，并较好地完成了与儿童用户之间的语言交流。另外，还较为广泛地抽样调查了儿童对于不同形态的机器人的认知方式，并得到了一些有意义的结果^[10]。

韩国高级科学与技术研究院与美国汉森机器人公司合作研制的 HUBO 机器人，具有逼真的人脸、身高 137cm、体重 57kg，全身具有 66 个自由度，通过非线性状态空间的统计学模型及表情合成技术，可以表达出喜悦、愤怒、悲伤等几种典型的表情^[12]，并在相应展览会上，与儿童进行有关科普方面的交流，取得了较好效果。

国内对这项技术的研究起步较晚，我国 973、863、“十五”计划均将人机交互列为主要內容。中国科学院软件研究所人机交互方法及智能信息处理实验室是最早开始多模态人机交互研究和开发的单位之一，分别承担了国家自然科学基金资助的重点项目“自然、高效和面向主流的多通道（模态）用户界面研究”，国家重点基础研究发展规划（973）“虚拟现实的基础理论、算法及其实现”中的子课题“自然、和谐的人机交互理论和方法”，国家 863 高科技发展计划“基于笔和语音的多模式融合的人机交互技术与应用”“基于多功能笔式交互的用户界面”“基于 PC 的多通道人-机交互开发环境”“虚拟现实中的三维交互技术”等，取得了重要的研究成果，并申请了相关的国家专利。

中国科学院软件研究所和北京大学计算机系、杭州大学工业心理学国家专业实验室合作承担的国家自然科学基金重点项目“多通道用户界面研究”，在我国首次对智能人机界面中多通道交互方式进行了系统性的研究，对多通道用户界面的模型、描述方法及整合算法、多通道用户界面开发环境、多通道用户界面的评估等方面都进行了一定的探索，取得了基础性的研究成果。清华大学计算机系、中科院自动化所、北京大学计算机系、中科院计算所、北京科技大学等单位在智能空间、笔输入用户界面、自然语言交互等方向也都做了大量工作。中科院心理所、浙江大学心理系、北京师范大学心理系，也从认知心理学角度对用户界面进行了研究，取得了许多重要的研究成果。汉王科技推出了具有自主知识产权的手写汉字识别系统。

从技术角度来看，研究的内容已遍及如生物特征信息提取、多模态信息采集、数据融合、信息处理、机电系统协同控制、软硬件协同设计技术、机电系统联合振荡的抑制等诸多领域。

由于人类五种主要的感觉（视觉、听觉、嗅觉、味觉、触觉）中，视觉所接收的信息

量居于首位，人体中 70% 的感觉集中来自眼睛，因此视频图像的采集、特征提取、自动辨识和理解对交互技术的研究起着重要的作用。语音技术同样在交互过程中起着不可忽视的作用，主要包括语音交互内容的研究和语音情感分析技术的研究。

在人与机器人和谐共存理论研究及应用技术领域，国内目前比较有代表性的是哈尔滨工业大学研究的多功能感知机，将人工智能与并行处理相结合，将智能体技术、数字模拟混合计算技术、并行计算技术、实时处理技术以及语音识别、表情识别、人脸识别、人脸检测与跟踪、文字识别、手语识别、手语合成、表情合成、自然语言理解等技术有机结合在一起，构造一个可以研究和开发包括视觉、听觉等人类语言的智能机器人综合交互平台^[13]；此外，清华大学以认知心理学为基础，对基于人工情感的拟人机器人控制体系结构进行了研究，同时根据其控制结构具有混合分层的特点，研究了典型认知情感状态下，机器人与人类用户交流中的整个信息处理过程的机制与方法^[14]。

2.2.3.1 人机接口方法的改进

伴随着计算机技术的飞速发展，人机接口技术也在不断改进：从早期的穿孔纸带、面板开关和显示灯等交互装置，发展到今天的视线追踪、语音识别、感觉反馈等具有多种感知能力的交互装置。用户界面的发展历经了批处理、命令行、图形界面三个阶段，现在的研究和开发重点已经放在了 Post-WIMP 界面上。

1. 批处理界面

在计算机发展的初期，人们通过批处理的方式使用计算机，这一阶段的用户界面是使用穿孔卡片作为输入设备，行式打印机作为输出设备。这只是用户界面的雏形阶段。

2. 命令行界面

在 20 世纪 60 年代中期出现的交互终端和分时系统中，已经开始考虑如何提供给用户方便实用的界面，这些系统提供了问答式对话、文本菜单或者命令语言进行交互，这个时期的人机界面称为命令行界面（Command Line Interface，CLI）。尽管熟练掌握命令语言后，人们能够灵活高效地操纵计算机，但是人们通常需要对语言进行大量记忆，在使用中很容易产生错误。

3. 图形用户界面

从 20 世纪 60 年代开始，由于超大规模集成电路的发展、高分辨率显示器和鼠标的出现，人机界面进入了图形用户界面（Graphical User Interface，GUI）的时代。图形用户界面的主要特点是桌面隐喻、WIMP（Window Icon Menu Pointing Device，窗口、图标、菜单、指点设备）技术、直接操纵和所见即所得。

桌面隐喻：界面隐喻（Metaphor）是指用现实世界中已经存在的事物为蓝本，对界面组织和交互方式的比拟。将人们对这些事物的知识（如与这些事物进行交互的技能）运用到人机界面中来，从而减少用户必需的认知努力。桌面隐喻采用办公的桌面作为蓝本，把图标放置在屏幕上，用户不用键入命令，只需要用鼠标选择图标就能调出一个菜单，用户可以选择想要的选项。

WIMP 技术：WIMP 界面可以看做是命令行界面后的第二代人机界面，是基于图形方式的。WIMP 界面蕴含了语言和文化无关性，并提高了视觉搜索效率，通过菜单、小装饰（Widget）等提供了更丰富的表现形式。

直接操纵：直接操纵用户界面（Direct Manipulation User Interface）是 Schneiderman 在

1983 年提出来的，特点是对象可视化、语法极小化和快速语义反馈。在直接操纵形式下，用户是动作的指挥者，处于控制地位，从而在人机交互过程中获得完全掌握和控制权，同时系统对于用户操作的响应也是可预见的。

所见即所得 (What you see is what you get, WYSIWYG)：也称为可视化操作，使人们可以在屏幕上直接正确地得到即将打印到纸张上的效果。所见即所得向用户提供了无差异的屏幕显示和打印结果。

2.2.3.2 人机接口方法的发展趋势

随着计算机技术和人工智能技术的飞速发展，可用于双足步行机器人的交互技术也会蓬勃地发展。

1. WIMP 界面发展

多媒体计算机和虚拟现实 (VR) 系统的出现，改变了人与计算机通信的方式和要求，使人机交互发生了很大的变化。在多媒体系统中继续采用 WIMP 界面有其内在的缺陷：随着多媒体软、硬件技术的发展，在人机交互界面中计算机可以使用多种媒体，而用户只能同时用一个交互通道进行交互，因而从计算机到用户的通信带宽要比从用户到计算机的大得多，这是一种不平衡的人机交互。

基于 WIMP 技术的图形用户界面，从本质上讲，是一种二维交互技术，不具有三维直接操作的能力。要从根本上改变这种不平衡的通信，人机交互技术的发展必须适应从精确交互向非精确交互、从单通道交互向多通道交互以及从二维交互向三维交互的转变，发展用户与计算机之间快速、低耗的多通道界面。

可以看出在计算机系统不同的发展阶段中，人机交互模型的发展过程如图 2-1 所示。在传统的人机系统中，人被认为是操作员，只是对机器进行操作，而无真正的交互活动。在计算机系统中人还是被称为用户。只有在 VR 系统中的人才是主动的参与者。

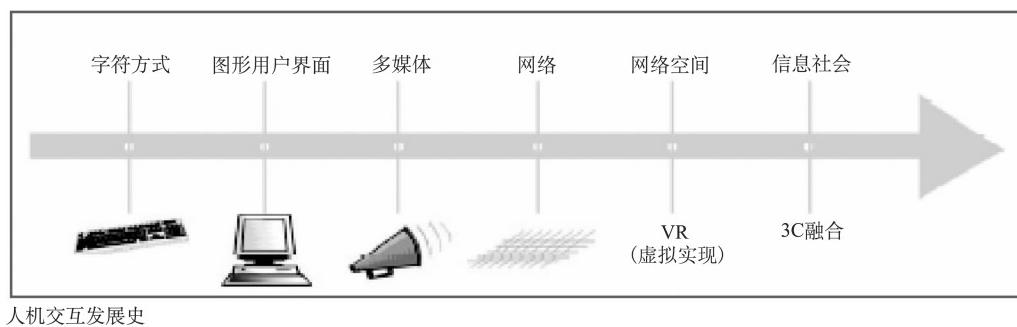


图 2-1 人机交互模型的发展过程

2. 多媒体与虚拟现实系统的交互

与传统用户界面相比，引入了视频和音频之后的多媒体用户界面，最重要的变化就是界面不再是一个静态界面，而是一个与时间有关的时变媒体界面。人类使用语言和其他时变媒体（如姿势）的方式完全不同于其他媒体。从向用户呈现的信息来讲，时变媒体主要是顺序呈现的，而我们熟悉的视觉媒体（文本和图形）通常是同时呈现的。在传统的静止界面中，用户或是从一系列选项中进行选择（明确的界面通信成分），或是用可再

认的方式进行交互（隐含的界面通信成分）。在时变媒体的用户界面中，所有选项和文件必须顺序呈现。由于媒体带宽和人的注意力的限制，在时变媒体中，用户不仅要控制呈现信息的内容，也必须控制何时呈现和如何呈现。

目前，许多人把多媒体系统错误地只当作是一种表现装置。这除了对多媒体的错误理解外，没有有效的多媒体交互形式也是目前多媒体存在的一大问题，因而多通道与多媒体用户界面是联系在一起的。

3. VR 系统中人机交互

人机交互可以说是 VR 系统的核心，因而，VR 系统中人机交互的特点是所有软、硬件设计的基础。其特点如下：

- 1) 观察点 (Viewpoint) 是用户做观察的起点。
- 2) 导航 (Navigation) 是指用户改变观察点的能力。
- 3) 操作 (Manipulation) 是指用户对其周围对象起作用的能力。
- 4) 临境 (Immersion) 是指用户身临其境的感觉，这在 VR 系统中越来越重要。

VR 系统中人机交互若要具备这些特点，就需要发展新的交互装置，其中包括三维空间定位装置、语言理解、视觉跟踪、头部跟踪和姿势识别等。多媒体与 VR 系统的人机交互有着某些共同特点。首先，它们都是使用多个感觉通道，如视觉和听觉；其次，它们都是时变媒体。

4. 多通道人机交互方法

人机交互技术是目前用户界面研究中发展得最快的领域之一，对此，各国都十分重视。在美国国防关键技术中，人机界面不仅是软件技术中的重要内容之一，而且是与计算机和软件技术并列的 11 项关键技术之一。欧盟的欧洲信息技术研究与发展战略计划 (ESPRIT) 还专门设立了用户界面技术项目，其中包括多通道人机交互界面 (MultiModal Interface for Man-Machine Interface)。保持在这一领域中的领先，对整个智能计算机系统是至关重要的。我们可以以发展新的人机界面交互技术为基础，带动和引导相关的软、硬件技术的发展，使更有效地使用计算机的计算处理能力成为可能^[15]。

21 世纪，人机交互的特点是多通道、多媒体。以虚拟现实为代表的计算机系统的拟人化，以及以手持电脑、智能手机为代表的计算机的微型化、随身化、嵌入化，是当前计算机的两个重要的发展趋势。通俗地讲就是，手机越来越大、电脑越来越小。鼠标和键盘为代表的 GUI 技术不再是主导，而是利用人的多种感觉和动作通道（如语音、手写、姿势、视线、表情等输入），以并行、非精确的方式与（可见或不可见的）计算机环境进行交互，大大提高了人机交互的自然性和高效性。

5. 语音交互方法

让计算机能听、能看、能说、能感觉是未来人机交互的发展方向，其中语音成为未来最被看好的人机交互方式，语音比起其他的交互方式有更多的优势。最早的语音技术因“自动翻译电话”计划而起，包含了语音识别、自然语言理解和语音合成三项非常重要的技术。语音识别的研究工作可以追溯到 20 世纪 50 年代 AT&T 贝尔实验室的 Audry 系统，此后研究者们逐步突破了大词汇量、连续语音和非特定人这三大障碍。20 世纪 90 年代前期，许多著名的大公司，如 IBM、苹果、AT&T 和 NTT 都对语音识别系统的实用化研究投以巨资。让计算机说话需要用到语音合成技术，其核心是文语转换技术 (Text to Speech)，语音合成甚至

已经应用到汽车的信息系统上，车主可以将下载到系统计算机中的文本文件、电子邮件、网络新闻或小说转换成语音在车内收听。还可以通过特定的服务网站订阅新闻，享受个性化的网络信息服务。2008年是语音技术重要应用的一年，“数字奥运”的目标就是研发“面向奥运多语言信息服务系统”，多语种语音合成产品是构成整个服务网络系统的重要组成部分，这套系统可以提供多语种、多方言、多发言人、多语气语调，并且具备一定自学习能力的个性化语音服务。

6. 眼标交互方法

交互基本上都离不开用户的视觉，那么如果用户能用眼睛（直接用大脑思维来控制的“脑标”目前还太遥远）来操控图形界面岂不是更方便？由此，“眼标”应运而生。2002年伦敦帝国学院的科研小组花费数年时间来研究人类眼睛活动与大脑感知之间的关系，最终发明出一种专门用来捕捉眼球细微动作的红外线感应装置，被他们称为“眼标”。“眼标”能鉴别眼睛在显示屏上的移动和注视，如果电脑使用者盯住屏幕上的某个链接图标1s以上，电脑就自动打开该链接；如果使用者盯住打印或者存盘的图标，电脑就会自动进行相应的操作。不过，人们眼球运动时存在的固有抖动以及眼睛眨动所造成的数据中断，就会干扰信号。另外，如果鼠标器光标总是随着用户的视线移动，可能会引起反感，因为用户“随便”看着什么而并非总是“意味着”什么。理想情况是用户希望发出控制时，系统及时地处理其视线输入，而在相反的情况下则忽略其视线的移动。让电脑去理解用户的这一美好愿望还不大可能。不过，科研人员仍认为人类用眼睛搜索和盯住一个目标的准确性远远超过用手移动鼠标。

2.3 理论难点

双足步行机器人是一种具有人的外形，并能够效仿人体的某些物理功能、感知功能及社交能力并能承袭人类的部分经验的机器人。因此人类研究双足步行机器人的目的是要创造一种新型工具，它能在典型的日常环境中和人类交流，在更广泛的环境任务中扩展人类的能力，因此研究双足步行机器人就会遇到许多的理论难点，如人工智能、人工心理与人工情感、遗传算法的应用等，只有攻克这些难题之后才能更好地研究机器人，造福人类。

2.3.1 人工智能理论

机器人的实际控制过程中，我们看到最多的应用还是PI、PD和PID控制。传统控制（经典控制理论和现代控制理论）的主要特征是基于模型的控制，即首先要建模——实现对实际被控物理对象的数学模型的建立。但是随着人类生产力发展水平的提高，需要被控制的对象越来越复杂。其复杂性表现为高度的非线性、高噪声干扰、动态突变性以及分散的传感元件和执行元件、分层和分散的决策机构、多时间尺度、复杂的信息结构等。这些复杂性都难以用精确的数学模型（微分方程或差分方程）来描述。然而，对这样复杂系统的控制性能的要求越来越高，使得基于精确模型的传统控制系统难以解决。因此把人工智能的方法引入控制系统，将控制理论的分析和理论的洞察力与人工智能的灵活框架结合起来，才有可能得到新的认识上的突破。智能控制系统具有拟人的智能或仿人的智能，即人工智能。这种智能主要表现在智能决策上，能够有效地解决复杂性和不确定性的控制问题。

模糊控制就是在研究人的控制行为特点的基础上发展起来的。对于无法构造数学模型的被控制对象，让计算机模仿人的思维方式，进行控制决策。人的控制可以用语言加以描述，总结成一系列的条件语句，即控制规则。运用微机的程序来实现这些控制规则，这样就很像是人的思考行为了。总之，模糊控制是基于专家经验和专业领域的知识，总结出若干条模糊控制规则，构成描述具有不确定性复杂对象的模糊关系，通过被控系统输出误差和模糊关系的推理合成为获得控制量，从而对系统进行控制。

模拟人类大脑的机能，人们又创造出了人工神经网络（可以通过计算机软件实现或通过大规模集成化硬件电路实现），并且进一步实现了神经网络控制系统，即在控制系统中采用神经网络这一工具对难以精确描述的、复杂的、非线性对象进行建模或充当控制器，或优化计算，或进行推理、故障诊断等。神经网络控制具有很强的逼近非线性函数的能力，即非线性映射能力。神经网络还具有自学能力、自适应能力，可以实现最优化的决策控制、神经网络学习控制、自组织控制。将神经网络控制与模糊控制相结合，可以实现更加复杂高效的神经网络模糊控制系统。在人工智能的新技术不断出现及智能控制的应用不断深化的过程中，神经网络必将在和其他的新技术相融合中发挥出更大的作用。

2.3.2 人工心理与人工情感

人工智能发展的最高水平的体现就是现代智能机器人。智能机器人要实现更为复杂的仿人控制，就不仅要使它具有人类的智力思维模式，还需要使机器人具有人类的情感思维模式。研究表明，人类的思维决策是辩证逻辑思维与情感思维的结合。情感思维可以使个体以最佳的行为方式（即控制方式），与社会的其他成员，特别是与其具有情感联系的某些人之间协同交互作用，达到最优化的效果。在未来的智能机器人控制系统中，不具备情感思维模式的控制是难以想象的，是不完备的。这就要求在机器人身上实现仿人类的人工感情思维控制。这一问题，牵涉到机器人与人类之间的情感交流以及机器人相互之间的情感交流。

人工智能的研究发展已经到了较高的水平，同时它的研究内容也在逐渐扩展和延伸。新世纪科学的研究发展的特征是多学科交叉，而 NBIC 会聚技术为人工智能的研究指明了方向。研究情感与认知的关系需要人工智能领域的专家扩展研究人工情感与人工心理问题。要使计算机拥有人工情感乃至人工心理处理能力，这一观点首先是由人工智能创始人之一的美国麻省理工学院人工智能学权威明斯基（Minsky）教授提出的，早在 1985 年他就指出：问题不在于智能机器能否拥有任何情感，而在于机器实现智能时怎么能够没有情感。目前对这个领域的研究已经在国内许多单位展开，可以说，对人工情感乃至人工心理的研究已经成为一种趋势。能够识别人类的情感，能够了解和模仿人类的心理，只有这样才能为人类提供更温馨的服务，即具有人工心理和人工情感的机器才具有更加友好的人机交互功能。

2.3.3 遗传算法的应用

近年来遗传算法得到迅速发展。遗传算法基于对生物遗传和进化过程的模拟，使得人工系统具有良好的自适应能力和优化能力。遗传算法提供了一种求解复杂系统优化问题的常用方法。因此将遗传算法应用于机器人这类复杂的人工系统是顺理成章的事情。它可以充分利用计算机的强大计算能力对解空间进行搜索。

遗传算法（Genetic Algorithm，GA），是以遗传学理论基础所构造的一类搜索算法，它

在某种程度上对生物进化过程进行了数学方式的模拟。遗传算法之父荷兰德 Holland 教授在他的著作《自然和人工系统中的适应性》(The Adaptability of Natural and Artificial System) 中将遗传算法的基本框架分解为适应系统和进化系统两个方面。

作为一种可用于复杂系统优化计算的鲁棒搜索算法，遗传算法不同于其他一些优化算法，具有以下一些特点：

(1) 对先验性知识依赖程度很低 主要通过适应度来判断所得候选解的效果，避免了比较复杂的建模等环节，在一定程度上具有某种非程序的特点和高度的自适应信息处理能力。

(2) 遗传算法以决策变量的编码作为运算对象 传统的优化算法往往直接利用决策变量的实际值本身来进行优化计算，而遗传算法以决策变量值的某种形式的编码为运算对象。这种对决策变量的编码处理方式，使得我们在优化计算过程中可以借鉴生物学中染色体和基因等概念，可以模仿自然界中生物的遗传和进化等机理，也使得我们可以方便地应用遗传操作算子。特别是对一些无数值概念或很难有数值概念，只有代码概念的优化问题，编码处理方式更显示出其优越性。

(3) 遗传算法直接以目标函数值作为搜索信息 传统的优化算法不仅需要利用目标函数值，而且往往需要目标函数的导数值等其他一些辅助信息才能确定搜索方向。而遗传算法仅使用由目标函数值变换得到的适应度函数值，就可确定进一步的搜索方向和搜索范围。这个特性是非常有用的，对那些目标函数无法求导数的函数，或导数不存在的函数优化问题及组合优化问题等，应用遗传算法时就显得比较方便，因为它避开了函数求导这个障碍。再者，直接利用目标函数值或个体适应度，也使我们可以把搜索范围集中到适应度较高的部分搜索空间中，从而提高了搜索效率。

(4) 遗传算法同时使用多个搜索点的搜索信息 传统的优化算法往往是从解空间中的一个初始点开始最优解的迭代搜索过程。单个搜索点所提供的搜索信息毕竟不多，所以搜索效率不高，有时甚至使搜索过程陷于局部最优解。遗传算法是从很多个体所组成的一个初始群体开始搜索最优解，而不是从单一的个体开始搜索。

(5) 遗传算法使用概率搜索技术 很多传统的优化算法往往使用的是确定性的搜索方法，在这种算法中一个搜索点到另一个搜索点的转移有确定的转移方法和转移关系，这种确定性往往也有可能使得搜索永远达不到最优点，因而也限制了算法的应用范围。而遗传算法属于一种自适应概率搜索技术，其选择、交叉、变异等运算都是以一种概率的方式来进行的，从而增加了其搜索过程的灵活性。虽然这种概率特性也会使群体中产生一些适应度不高的个体，但随着进化过程的进行，新的群体中总会更多地产生出优良的个体，实践和理论都已证明了在一定条件下遗传算法总是以概率 1 收敛于问题的最优解。当然，交叉概率和变异概率等参数也会影响算法的搜索效果，所以如何选择遗传算法的参数在具体应用时是一个比较重要的问题。

随着计算机计算能力的不断提高，遗传算法在机器人控制领域也得到了应用。

2.3.4 基于仿生学原理的步态控制

基于仿生学的双足步态研究主要通过测量和分析人类的步行运动，研究双足步行的基本原理，将得到的一些基本步态特征运用到双足步行机器人控制当中。最早在 1901 年英国摄影家爱德沃德迈布里奇 (Eadweard Muybridge) 通过观察人类和 40 多种动物的运动过程，研

究了包括马等 8 种动物的运动步法，得到了它们的步态模式。他的研究揭示了动物的简单步态模型，开辟了从仿生学角度研究双足步态的先河。随后将行为模仿的思想运用到机器人 VR 的控制方法当中，通过模仿学习表达非线性动力学差分方程，形成控制策略，采用的局部权重回归算法保证了学习的收敛性。沿着这个思路，1976 年 Margaria 通过研究人类不同步行运动过程中肌肉的电信号和能量消耗，探讨了人类步行所需要的最少肌肉运动量，这为以后研究被动步态提供了一定的事实依据。随后美国南加州大学的 Auke Ijspeert 博士等将行为模仿的思想运用到了机器人 VR 的控制方法当中。通过模仿学习表达非线性动力学差分方程形成控制策略，采用的局部全重回归算法保证了学习的收敛性。这种研究思路也被日本本田公司运用到了 ASIMO 机器人的步态规划当中，通过离线研究人类步态运动的特性，形成简单的步态模式，然后经过在线调整误差控制机器人最终步态。

另一个仿生学的重要贡献是在大多数动物脑组织中发现了控制运动节奏的中央模式发生单元（Central Pattern Generator），这促使一部分研究学者尝试在双足步态中应用这一结构。1999 年 Williamson 用中央模式发生器控制机器人手臂运动完成简单的杂要动作，这位学者还从人类步行机构得到启示，设计弹性关节解决双足步行中关节冲击问题。Fukuoka 等设计了 PD 控制器组成的弹性关节，通过中央模式发生单元接受传感器输入，并将计算得到的输出角度发送给各运动关节。实验表明机器人可以在中度不平整地面行走，并可根据地面平滑程度调整能量消耗和稳定性两项运动指标。他们的成功吸引了越来越多的学者开始这一方向的研究。

人体步行运动的研究为实现机器人步行运动提供了借鉴和指导，对人类步行运动的观察与记录也促使人们了解和发现步行的运动特征与规律。在获得人类行走数据的基础上，根据仿生学思想，研究人员还尝试使用记录的人类步行运动数据驱动机器人行走。目前世界上最先进的自主式拟人机器人 ASIMO 就是在分析人类行走运动的基础上动态生成双足步态。随着测试和记录手段的不断完善，研究人员可以获得更加完整和详细的步行运动数据，由此提炼得到更加完备的人体运动规律，设计更加接近人类步态的双足机器人步态。

2.3.5 动力学模型

机器人学学术界对双足运动的研究已有多年，其中比较好的入门读物有前南斯拉夫学者 Vukobratovic 与美国麻省理工学院的 Raibert 的著作。

在过去的 40 多年里，公司、大学、研究所开发的双足机器人样机数以百计，在这其中，日本本田公司在样机技术上取得的进步最让人瞩目，但从其公开的录像可以看出，本田机器人 ASIMO 仍然采用零力矩点（Zero Movement Point, ZMP）稳定判据规划其运动，在相同的步速下，与人类的行走模式还有较大区别，且无法解释和实现 ZMP 稳定判据以外的双足运动。正如美国麻省理工学院腿实验室的 Pratt 教授所说：本田公司设计控制器的方法无法解释为什么给定的轨迹有效，也无法解释在崎岖路面上选择轨迹模式的内在机理。

双足步行机器人要取得进一步的发展，就必须要研究双足运动的内在固有特点，必须深入研究其稳定性与控制机理。只有解释清楚了这一系列机理性问题，双足步行机器人的研制才能有本质上的进步。

双足行走机器人与多足行走机器人相比，本质上是不稳定的，而且控制更加困难。人类以其完美的行走姿态和稳定性成为双足步行机器人的模仿对象。由于人类的行走与机器人的

行走机理不同，完全生搬硬套地将人类行走的模型应用到双足步行机器人上显然是不行的。因此研究稳定的姿态与步态一直是各国研究双足机器人的最重要的问题之一。

2.3.5.1 动力学分析

双足步行机器人是一个多自由度的复杂机械结构，其动力学方程具有非线性、高阶和强耦合的特点，现有的数学模型和解析算法不足以得到完整和精确的步态解。除可控的有驱动自由度以外，双足系统脚板与地面之间存在无驱动的自由度，这个自由度的控制对机器人的姿态稳定具有重要的意义。最后双足步行机器人不同特性的步态周期切换也是自然步态生成中的一个难点。双足步行机器人的步态由静态的双脚支撑期和动态的单脚支撑期组成。在步行过程中，双足步行机器人从稳定的双脚支撑过渡到不稳定的单脚支撑，相应的控制结构从闭环切换到开环。从机器人的跑步运动我们可以看到，机器人在腾空状态依然可以保持动态平衡，也就是说机器人的步行控制不能平行地移植到机器人的跑步控制当中。这为多模型系统的稳定控制和平滑控制制造了更多的困难。

双足步行机器人是一个多自由度、多关节的复杂多连杆机构，具有丰富的动力学特性。对双足步行机器人进行动力学分析可以帮助了解行走运动和力之间的关系，明确机械本体的质量特性、步行姿态、步行速度与关节力矩的相互联系，为对其进行力矩控制和平稳性分析提供了依据。为便于分析，需要对其进行动力学建模并进行简化。

双足步行机器人分为前向模型和侧向模型。前向模型主要用来分析行走前行时的步态特性以及步行时的平稳特性，而侧向模型则更多地体现了行走时侧向的平稳性。国内外学者曾对两足步行机构的前向模型和侧向模型进行研究和假设性分析，如马宏绪等人在一定假设的基础上认为双足步行机构的侧向和前向的步行运动互不影响，可以分开建模分析，以便得到用于步态设计和控制规律综合的双足步行机器人模型^[16]。正是借鉴于前人研究的建模法，对双足步行机器人的前向模型和侧向模型分开建模。

迄今为止，对双足步行机器人的基本研究思路可以分为基于仿生学原理和基于动态控制原理两种类型。这两种不同思路的研究方法在双足步行机器人的步态设计和规划中都有广泛应用。

2.3.5.2 ZMP 稳定性判据

双足步行机器人研究的一个关键问题是实现其稳定地行走。目前世界上大多数双足步行机器人系统都采用 ZMP 作为稳定行走的判据。ZMP 即零力矩点，是由前南斯拉夫学者 Vukobratov（简称伍氏）提出的关于步行稳定性的经典理论，他研究了 ZMP 与双足动态系统之间的关系，提出 ZMP 是判断动态平衡的一个重要依据。ZMP 是这样描述的：ZMP 是地面上的一点，重力和惯性力对这一点的力矩，其水平分量为零。也就是说整个系统对于这个点的前向、侧向的倾翻力矩为零。他指出，当双足机构处于动态平衡时，ZMP 和脚底所受地面反力的压力中心（Center of Pressure, CoP）是重合的。因此我们可以根据检测到的地面对力信息，计算 CoP，通过控制策略调整 ZMP 和 CoP 的位置，使二者重合，实现机器人的动态稳定行走。

作用在机器人脚底的实际地面反力（包括垂直反力和摩擦力）作用点如能与期望 ZMP 重合，并落在支撑面上，则对于机器人无翻转力矩，从而使机器人处于稳定行走状态。

期望 ZMP 计算公式

$$\begin{aligned}
 X_{\text{ZMP}} &= \frac{\sum_{i=1}^n [m_i \ddot{x}_i (\ddot{z}_i + g) - m_i z_i \ddot{x}_i + V_{iy}]}{\sum_{i=1}^n m_i (\ddot{z}_i + g)} \\
 Y_{\text{ZMP}} &= \frac{\sum_{i=1}^n [m_i \ddot{y}_i (\ddot{z}_i + g) - m_i z_i \ddot{y}_i + V_{ix}]}{\sum_{i=1}^n m_i (\ddot{z}_i + g)} \\
 Z_{\text{ZMP}} &= 0
 \end{aligned} \tag{2-1}$$

式中 x 、 y 、 z ——机器人身体各部分质心坐标；

m_i ——机器人身体各部分的重量；

V_{iy} 、 V_{ix} ——关节驱动力。

XOY 在脚底板所在平面， Z 轴垂直地面向上。

实际 ZMP 计算主要由六维力/力矩传感器进行测量获得。

机器人脚底的实际 ZMP 点和期望 ZMP 点的分布示意如图 2-2 所示。

如不重合，则翻转力矩计算式为

$$T_{\text{前向}} = (X_{\text{期望ZMP}} - X_{\text{实际ZMP}}) Z_{\text{向广义力}} \tag{2-2}$$

$$T_{\text{侧向}} = (Y_{\text{期望ZMP}} - Y_{\text{实际ZMP}}) Z_{\text{向广义力}} \tag{2-3}$$

$$\begin{aligned}
 T_{\text{旋转}} &= (X_{\text{期望ZMP}} - X_{\text{实际ZMP}}) Y_{\text{向广义力}} + \\
 &\quad (Y_{\text{期望ZMP}} - Y_{\text{实际ZMP}}) X_{\text{向广义力}}
 \end{aligned} \tag{2-4}$$

式中 $T_{\text{前向}}$ 、 $T_{\text{侧向}}$ 、 $T_{\text{旋转}}$ ——作用在机器人上的前向力矩、侧向力矩和旋转力矩；

$X_{\text{期望ZMP}}$ 、 $Y_{\text{期望ZMP}}$ ——期望 ZMP 点的坐标；

$X_{\text{实际ZMP}}$ 、 $Y_{\text{实际ZMP}}$ ——实际 ZMP 点的坐标。

根据 ZMP 原理得到的结论：

1) 通过改变脚的姿态，改变实际 ZMP 的位置；

2) 通过改变惯性力的大小（速度和姿态），改变期望 ZMP 的位置。

以 ZMP 作为双足步行机器人动态行走的稳定判据，建立了基于六维力/力矩传感器的实际 ZMP 检测系统，使双足步行机器人能够实现 ZMP 的闭环控制。

ZMP 本是针对双足步行提出的，但一些研究人员将其应用到双足跑步领域。Kajita 将机器人简化为一个无质量弹簧腿的倒立摆，在单腿支撑期遵守 ZMP 准则，在腾空期根据弹道模型得到飞行轨迹。国内有人将跑步稳定性分为支撑阶段和腾空阶段，在支撑阶段限制机器人足与地面完全接触，仍然遵守 ZMP 稳定准则。2003 年，日本索尼公司发布了其跑步机器人 QRIO 的实验录像。2004 年底，日本本田公司公布了 ASIMO 的跑步实验录像。基于商业机密的考虑，以上公司没有公布其技术细节，但从录像可以明显看出，这些机器人平脚板落地和平脚板腾空，仍然遵守 ZMP 平衡准则。实质上，ZMP 的基本概念和判据是通过对约束的描述，确切地说是对机器人足与地面完全接触的约束，来间接反映静态和准动态步行的平衡性，由于没有从步行的本质去研究稳定性，并缩小了步行运动的范围，因而无法解释双足机器人行走时的动态运动。由于长期沿用 ZMP 稳定判据来研究双足步行机器人，各种相关

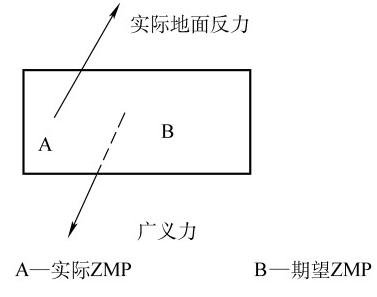


图 2-2 机器人脚底的实际 ZMP 点和期望 ZMP 点的分布示意图

研究一直没有从根本上突破其基本概念、定义及其理论模型，致使双足步行的基础理论严重滞后于拟人机器人样机技术的发展，并且已经成为进一步揭示人类复杂步行运动规律，提高样机步行性能，实现大步幅快速步行、动态跑步和跳跃等类人重要运动功能的瓶颈问题。

综上所述，近年来机器人领域的研究人员对双足步行机器人的稳定性与控制策略进行了大量的研究，取得了一系列的重要进展。但是在以下几个方面还需要更进一步研究：

(1) 双足运动的动态稳定与控制机理 稳定性是仿人机器人的核心问题，目前基于ZMP的准动态理论不能满足双足机器人今后的发展需要，也无法解释自然界双足生物的复杂运动。因此迫切需要一套具有更广应用范围的、适合双足机器人的动态运动的稳定性理论体系。

(2) 双足步行运动的固有鲁棒性机理 双足步行的鲁棒性问题是仅次于稳定性的另一个重要研究领域。双足生物对路面环境、自身的力学参数都具有很强的鲁棒性。而目前的双足机器人的步行运动对环境和自身的力学参数依赖性很大，因此给出这种鲁棒性的数学意义上的内蕴解释和严格证明就显得十分重要。

(3) 实时步态规划与控制 目前双足机器人步态规划的研究主要集中在离线的产生，这种规划模式在一定程度上限制了步行的灵活性。因此需要采用在线步态合成方法解决双足机器人对不同环境的适应性问题。计算机技术的发展在一定程度上解决了计算速度的瓶颈问题，因而实时步态规划与基于传感器的大闭环反馈控制是未来的研究方向。

2.3.6 人机交互的实现方法

在对交互过程中的多模式信息采集、认知特征提取、多模式行为关联融合，及人拟数字化分析方法与应用技术展开深入研究的过程中，围绕交互过程的多模式信息认知特征的提取、人拟数字化技术以及分布式认知模型的建立方法，构造符合人类感知和认知行为的典型应用背景下机器人个性化交互平台。具体实现方法包括以下6个方面：

(1) 用户基础信息的特征提取方法 该方法是寻求能够实时、准确地反映用户与其机器人交流过程中脸部表情、视线关注检测、身体姿态、声音及语言文字、肢体行为、步态信息等多模态特征的、可计算的表示方法，基于多模态信息融合技术，为人拟数字化特征表达及认知特征的自动获取，提供高效的基础性支持。

(2) 提取信息的人拟数字化特征表达方法 这种方法是研究高效、准确地表达交流过程中的人性化感知方法，基于交互过程中的脸部表情、身体姿态、语音信息、肢体行为等数据，构造出符合人类认知心理特征的智能化表达，消除“交互壁垒”问题。

(3) 多模态信息融合方法 该技术是基于图像、语音、文本、触觉及其他多种传感器所提取的可计算信息（已完成人拟数字化特征表达）的一种融合方法，可以增强基本特征的多源复合效用，提高机器人认知特征提取及交流过程的表达能力。

(4) 高效分布式认知特征的自动获取及其模型的建立方法 此方法研究如何基于人脑启发式的感知数字化特征表述，抽取能够恰当反映用户感知和认知信息的认知特征，以及如何对提取的认知特征进行有效评估的客观评价标准；以家庭背景下的机器人为具体研究对象，研究认知模型的建立方法，使得机器人的交流模式信息符合人类的感知与认知行为，产生的结果更加贴近用户的交流要求。此技术最终目的就是利用机器人数据处理方面的高精度、高鲁棒、高速度等计算特点对所提取的、大量的多模式信息进行符合用户行为的内容解析，得

到满足用户要求的表达结果，即在面向用户的机器人上建立有效的、用于模拟用户认知、反映用户交流过程中个性偏好的可计算特征。分布式认知模型的建立与引入，可以实现用户认知的可计算客观特征空间与其主观心理空间之间的映射。

(5) 关联融合的多模式行为合成方法 研究如何基于所获取的恰当反映用户感知和认知信息的认知特征以及分布式认知模型，研究符合其人因特征的多模态行为模式及其高效的关注融合算法和控制系统，使所产生的结果与交流过程中儿童生理和心理要求趋于一致。同时通过关注模型等手段，对于玩伴机器人关联融合的多模式行为合成进行客观的综合评估。

(6) 个性化实时交互方法 由于每个用户在情绪状态、性格特点、兴趣喜好等方面个体差异很大，对交流过程中的需求各式各样，因此这种技术结合了用户的认知特性来表达实现个性化的实时交互功能。利用此项技术可研制出融合分布式认知模型的、可进行个性化交互的机器人，使其能够根据用户的偏好需求，将基于人拟数字化特征表达的脸部表情、身体姿态、语音信息、肢体行为等模式自然地表现出来，完成不同交互模式的平滑切换，产生符合要求的个性化交互结果。根据在交互过程中视觉、语音及其他多传感器所获取的脸部表情、身体姿态、语言信息、肢体行为等信息，分析出其深层次的情绪状态、性格模式等个性化特征，去除掉用户厌烦的、不感兴趣的感知内容，提取出其真正关心的感知内容，由上述所建立的分布式认知模型以及个性化情感状态空间转移的非线性模型，实时自动合成出关联融合的多模式行为，以满足用户与机器人之间个性化的自然、和谐、舒畅的实时交流。

2.4 小结

本章主要介绍了双足机器人的相关理论与技术，双足步行机器人研究平台集多学科于一体，可涉及位置与姿态的稳定性控制、动力学建模、行为动作的自动生成、和谐人机交互、人工智能、人工心理与人工情感等多个方面。

参 考 文 献

- [1] Zeungnam Bien. Blend of Soft Computing Techniques for Effective Human-Machine Interaction in Service Robot Systems [J]. *Fuzzy Sets and Systems*, 2003 (134): 5-25.
- [2] Masahiro Fujita. Digital Creatures for Future Entertainment Robotics [C]. *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, San Francisco, CA, 2000, 4: 801-806.
- [3] Lola Canamero. Emotion Understanding from the Perspective of Autonomous Robots Research [J]. *Neural Networks*, 2005 (18): 445-455.
- [4] K Suzuki, Development of Face-to-face Communication Function for a Humanoid Robot [J]. *Systems and Computers in Japan*, 2006, 37 (14): 1-13.
- [5] H Kobayashi, F Hara. A Basic Study on Dynamic Control of Facial Expressions for Face Robot [C]. *Proceedings of the 1st International Workshop on Robots and Human Communication*, 1994: 186-195.
- [6] A S Nakaoka, K Ikeuchi. An Efficient Method for Composing Whole Body Motions of a Humanoid Robot [C]. *Proceedings of the 10th International Conference on Virtual Systems and Multimedia*, 2004, 11: 1142-1151.
- [7] Sarah Woods. Exploring the Design Space of Robots: Children's Perspectives [J]. *Interacting with Computers*

- puters, 2006 (18): 1390-1418.
- [8] Ronald C Arkin. An Ethological and Emotional Basis for Human-Robot Interaction [J]. Robotics and Autonomous Systems, 2003 (42): 191-201.
- [9] Stefan Sosnowski, Ansgar Bittermann. Design and Evaluation of Emotion-Display EDDIE, Proceedings of the 2006 IEEE/RSJ [C]. International Conference on Intelligent Robots and Systems, 2006, 10: 3113-3118.
- [10] Kidd C. Sociable Robot System for Real_World Problems [C]. Proceedings of the 14th IEEE Workshop on Robot and Human Interactive Communication, Nashville. 2005: 1023-1036.
- [11] Noriaki Mitsunaga. Robovie-IV: A Communication Robot Interacting with People Daily in an Office [C]. Proceedings of the International Conference on Intelligent Robots and Systems, 2006: 5066-5072.
- [12] Gemperle F, DisalvoC, Forlizzi J, et al. The hug: a new form communication [C] //Proceedings of the DUX 2003 conference. New York: ACM, 2003.
- [13] 徐琳, 袁保宗, 高文. 真实感人脸建模研究的进展与展望 [J]. 软件学报, 2003, 14 (4): 804-810.
- [14] 宋亦旭, 贾培发. 基于人工情绪的拟人机器人控制体系结构 [J]. 机器人, 2006, 26 (6): 491-495.
- [15] 王勇, 杨杰. 基于被动动力式的两足机器人研究现状 [J]. 机械工程师, 2005, 6.
- [16] 马宏绪. 双足步行机器人动态步行研究 [D]. 长沙: 国防科学技术大学, 1995.

第3章 双足步行机器人的相关技术

通过上一章的介绍和分析，我们不难发现制作双足机器人需要多种技术的支撑，其中主要技术包括：行为数据库技术、3D建模及虚拟现实技术、双足机器人建模技术、多智能体通信技术及多智能体协调控制技术等，下面具体加以介绍。

3.1 动作自动生成——行为数据库技术

我们所介绍的系统中，有一个重要的环节就是组建机器人行为数据库，机器人的行为和人的行为一样可以分为很多细节。一个连续的行为表达，是由很多个细小的行为元素组成的。下面以向前步行为例来说明行为数据库的组成。

机器人的行为数据库是由细分的行为数据单元组成的。每一个动作图片对应唯一的动作数据，这样的动作单元的不断积累，也就构成了我们的行为数据库，即行为发生器。图片的直观性也给我们组合其他复杂的动作带来了便利。

根据数据生成的原理可以保证最基本的动作数据的稳定性和可靠性。但是基本数据及其连接之间的过渡过程是否稳定、可靠就很难保证。为了解决这个问题，我们采用了两种办法：①采用静态数学模型稳定性判据来判断；②加一个过渡过程，即在一个基本动作的开头和结尾都加一个可靠的过渡点。实践证明，方法②较为适合。

双足步行机器人具有非线性、强耦合等特性，易受环境干扰，特别是机器人与地面接触环境的不确定性和局限性导致双足机器人行走控制难度增大。利用行为发生器，组合出了一组机器人上台阶的动作，并保证其稳定性。这也证明了本章所介绍的基于行为发生器的步态控制是可行的。当然，该方法也有很多不足之处有待改进，例如，通过提供大量的传感器反馈信息来提高控制精度。

3.2 3D 虚拟仿真技术

仿真是利用一个模型来模仿实际系统所发生的运动过程进行试验的技术，仿真技术在科学、工程设计、军事演练等领域中，早已广泛应用。现代计算机与仿真相结合，形成了计算机仿真技术，为仿真赋予新的内涵，并开辟了更加广阔的应用空间。计算机仿真技术作为一门新兴的高技术，在减少损失、节约经费、缩短开发周期、提高产品质量等方面发挥了巨大作用。

传统的产品开发过程一般分为功能设计、结构设计、样机研制与试验等几个阶段。其中样机制造与试验是传统产品开发过程中不可缺少的环节。虽然样机制造和试验可以解决许多在设计中无法预料的问题，但也带来了一个负面影响，即增加产品成本和延长了设计周期。随着计算机科学技术的发展和广泛运用，虚拟样机技术的出现解决了这个问题。它可以在计算机上方便地修改设计缺陷，仿真试验相同的方案，对整个系统进行不断改进，直至获得优

化设计方案以后，再出物理样机。

随着机器人技术的迅速发展，机器人在各方面也得到了广泛的应用。然而由于机器人综合了机、电、液的复杂动态系统，因此通过计算机仿真来模拟系统的动态特性，才能揭示机构的合理运动方案及有效的控制算法，从而解决在机器人设计、制造以及运行过程中的问题。机器人仿真技术大致分为以下几类：

1) 对制造系统中机器人的应用开展的研究，如柔性制造系统或计算机集成制造系统中机器人的仿真问题。

2) 针对机器人本身的特性进行仿真研究，如运动学仿真、动力学仿真、轨迹规划和碰撞检验等问题。

3) 机器人离线编程系统的研究，如利用仿真生成满意的运动方案自动转换成机器人控制程序去驱动控制其动作。机械工程中的虚拟样机技术又称为机械系统动态仿真技术，是20世纪80年代随着计算机发展起来的一项新技术，其核心是机械系统运动学和动力学仿真技术，同时还包括三维CAD建模技术、有限元分析技术、机电液控制技术、最优化技术等相关技术，见图3-1。运用虚拟样机技术，可以大大简化机械产品设计开发过程，大幅度缩短产品开发周期，大量减少产品开发费用和成本，明显提高产品质量，提高产品的系统性能，获得最优化和创新的设计产品。以计算机辅助设计为基础，把虚拟技术和仿真技术相结合，为产品的研发提供了一个全新的设计方法。这些技术构成了功能完善的虚拟样机系统的技术体系。包含技术简介如下：

(1) 三维建模的计算机辅助设计技术(CAD) 用于机械系统的几何建模，以及展现机械系统的仿真分析结果。

(2) 有限元分析(FEA)技术 可以利用机械系统的运动学和动力学分析结果，输出进行机械系统有限元分析所需的载荷和边界条件。同时也可利用有限元分析对构件的应力、应变和强度的结果，进行进一步的分析。

(3) 模拟各种作用力的软件编程技术 虚拟样机软件运用开放式的软件编程技术进行模拟，例如：液压冲击、气动力、风力、轮胎振动等，以满足各种机械系统的要求。

(4) 利用实验装置的实验结果进行构件的建模 实验结果经过线性化处理输入机械系统，成为机械系统模型的一个组成部分。

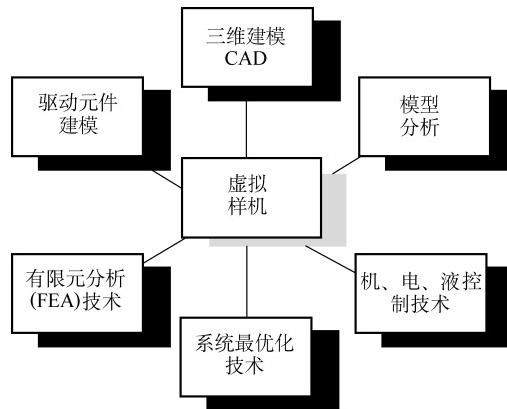


图3-1 虚拟样机系统

(5) 控制系统设计与分析技术 虚拟样机软件可以运用经典和现代控制理论，进行机械系统的运动仿真分析。或者可以应用其他专用的控制系统分析软件，进行机械系统和控制系统的联合分析。

(6) 优化设计和分析技术 运用虚拟样机分析技术进行机械系统的优化设计和分析，通过优化分析，确定最佳设计结构和参数值，使机械系统获得最佳的综合性能。

3.3 双足步行机器人建模的方法与技术

3.3.1 ADAMS 建模方法

机械系统动力学自动分析软件 ADAMS (Automatic Dynamic Analysis of Mechanical Systems)，是美国 MDI 公司开发的非常著名的虚拟样机分析软件。ADAMS 的整个计算过程指从数据的输入到结果的输出，不包括前、后处理功能模块。可以分成以下几个部分：①数据的输入；②数据的检查；③机构的装配及过约束的消除；④运动方程的自动形成；⑤积分迭代运算过程；⑥运算过程中的错误检查和信息输出；⑦结果的输出。

拟人机器人建立虚拟样机的物理模型是为了验证拟人机器人模型的运动性能，并通过模型计算出各个关节的驱动力矩、ZMP 的变化轨迹等。

3.3.2 模型的数据转换

在建立虚拟样机模型过程中，如果简单地通过 Mechanism/Pro 将已经设计完成的 Pro/E 模型数据转换到 ADAMS 中，在建立 ADAMS 模型时总是由于部分零件的缺失、装配关系错误导致整个模型无法正确显示。这是由于一些比较复杂的零件模型转入 ADAMS 后特征点和读取的路径缺失，从而造成数据传递只有质量、惯性等物理信息存在，而刚体却并不显示所造成的。

为了实现拟人机器人模型的数据转换，在对模型数据转换过程的反复观察和实践后发现，Mechanism/Pro 传递数据的实质是将装配体的各零件以 Stereolithography (.stl) 或 Render (.slp) 格式输出的同时，产生两个记录文件 transferToView.adm 和 aview.cmd。

TransferToView.adm 是 ADAMS 数据设置文件，它记录 ADAMS 的模型数据。主要内容为：装配体名、ADAMS 地面信息、ADAMS 零件位置信息（密度、转动惯量、MARKER 数量和位置）、单位制等。

aview.cmd 是 ADAMS 的命令文件，它记录 ADAMS 的操作命令。主要内容规定：ADAMS 的单位制（长度、质量、力、时间、角度、频率），读取模型的数据类型、路径、名称、比例、坐标、隶属关系，模型显示状态等。

这两个文件在传递过程中对零件在 ADAMS 中的生成和定义起了决定性的作用。如果在传递过程中将完整的零件特征、装配信息输入这两个文件中，就可以实现零件和模型的完整传递，如图 3-2 所示。

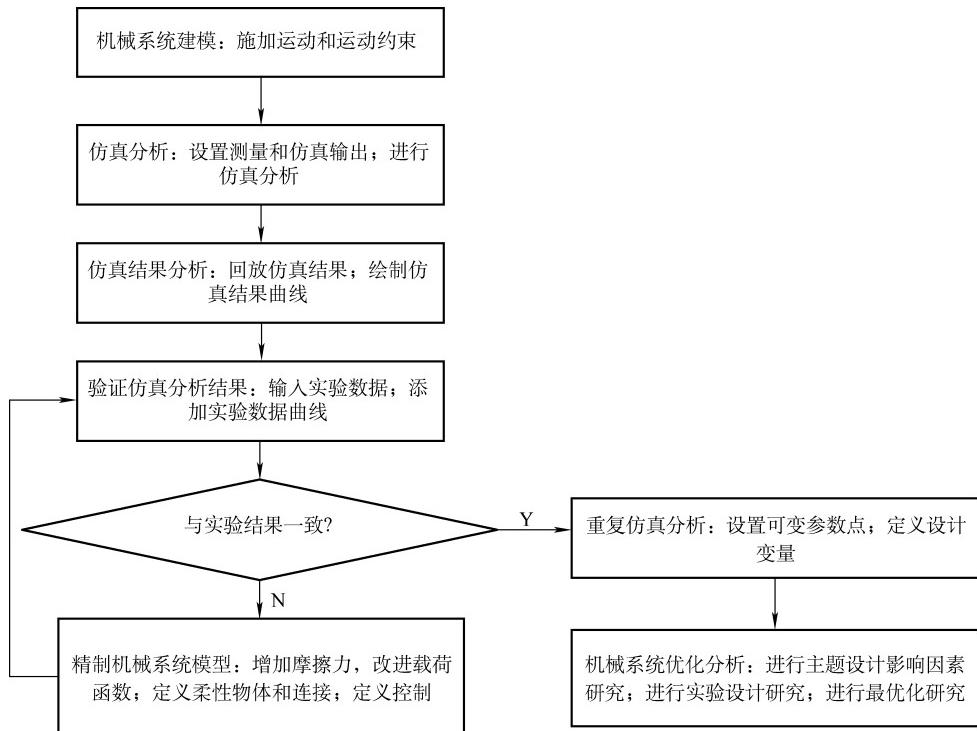


图 3-2 零件和模型的完整传递流程图

3.4 多智能体技术

智能体通过传感器感知环境，通过效应器作用环境，传感器相当于人的眼、耳、鼻及其他器官，效应器相当于手、腿、嘴、身体。智能体的工作过程如图 3-3 所示。智能体不仅要与环境交互，还需要对信息进行解释和处理。其中，信息融合以适当方式对接收到的信息进行处理。信息处理过程是智能体的核心，它反映了智能体的真正功能。

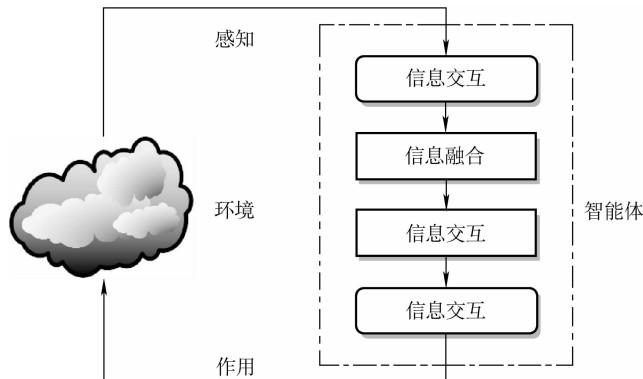


图 3-3 智能体的工作过程

智能体（Agent）是指那些宿主于复杂的动态环境，自主感知环境中信息，并采取行动，实现一系列预先设定的目标或任务的计算机系统。它具有推理和学习能力、自主与协同工作

能力、在所处环境中的灵活访问和迁移机制以及同其他智能体（Agent）的通信和协调机制，概括起来就是智能性、代理性和机动协调性。该技术最早出现在人工智能领域，其最大特点是具有一定的智能及良好的灵活性和坚定性，特别适合处理复杂、难以预测的问题。基于智能体的控制模型是一种拟人化的模型，可将控制系统的行爲和其他扩展单元的行爲统一起来构成分布式多智能体系统。

多智能体系统（Multi-Agent System）是由多个可计算的智能体组成的集合，它能协调一组自主体的行为（知识、目标、方法和规划等），协同完成动作^[1,2]。学习是多智能体系统的重要特征之一，多智能体学习可以看做是多智能体系统和机器学习等研究领域的结合。强化学习是一种以环境反馈作为输入的适应环境的机器学习方法。它不需要对环境的先验知识，具有无监督的自适应能力，因而被认为是设计智能体（Agent）的核心技术之一。

多智能体系统是针对一群不同种类的、自治的智能体的群体行为所进行的研究，这些智能体可能为同一目标共同工作，也可能为存在潜在冲突的各自不同的目标工作^[3]。这方面的研究主要受分布式人工智能理论和人工生命系统的影响，其研究的中心问题可以概述为一句话：“在一个只有我们所设计的智能体的世界中，智能体是如何与其他智能体进行相互作用的”。在具体的研究工作中，多个智能体之间如何协调运用各自的知识、目标、技能和计划共同采取行动、解决问题，是一个重要的研究内容。多智能体系统的应用分类如图 3-4 所示。从图中可以看到，多机器人系统的研究与多智能体系统研究是密切相关的。多机器人系统为多智能体系统理论应用于实际环境中提供了实验平台，同时多智能体系统理论的研究为多机器人系统和多机器人协调协作提供了理论基础^[4]。

近年来，随着机器人 Agent 模型在多机器人协调中的应用以及为了解决多机器人合作所存在的问题，基于分布式人工智能中多智能体系统（MAS）理论（见图 3-4）的多机器人协作已成为机器学研究领域的热点，受到各国专家学者的普遍关注。依据 MAS 的特性，组织和控制多个机器人，使之能合作完成单个机器人无法完成的复杂任务，其主要考察的是系统的智能行为，因此它不同于一般的控制系统而必须具有实时智能^[5]。

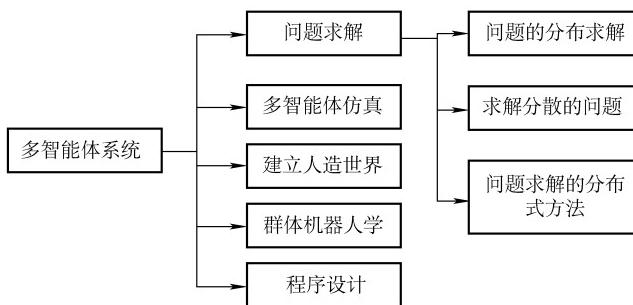


图 3-4 多智能体系统应用分类

3.4.1 智能体通信技术

3.4.1.1 通信方式

智能体的通信方式分为黑板系统和消息系统两种类型。在黑板通信系统中，智能体通过黑板交换信息、数据和知识，在黑板系统中智能体间不发生直接通信，采用消息通信可以实现灵活复杂的协调策略。

与黑板系统不同，两个智能体间消息是直接交换，执行中没有缓冲。广播是一种特例，消息发给每个智能体或智能体组。通常发送者要为消息指定唯一的地址，只有该地址的智能体才能读这条消息。为了实现协作，所有智能体必须知道语言的语义。

3.4.1.2 通信语言

目前国际上最著名的智能体通信语言是美国国防部高级研究计划署（Advanced Research Project Agency, ARPA）的知识共享计划（Knowledge Sharing Effort）中提出的两个相关语言：一个是 KQML（Knowledge Query and Manipulation Language），另一个是 KIF（Knowledge Interchange Format）。其中 KQML 是一种最通用的智能体通信语言。根据 KQML 规范，设计了一种适用于 MAPS（Manage Access Points）环境下的智能体通信语言（Software Agent Communication Language, SACL）。该语言的适用前提如下：

- 1) 智能体单方向通信；
- 2) 消息传送可以有延迟；
- 3) 消息接收方知道发送方；
- 4) 消息交互有序、可靠。

SACL 语言的 BNF 语法定义如下：

| | |
|-----------|--------------------------|
| <消息>:: | = <动作> (<空白> <参数—值对>) |
| <动作>:: | = <标志符> |
| <参数—值对>:: | = <参数名> <空白> <参数值> |
| <参数名>:: | = <标志符> |
| <参数值>:: | = (<ASCII 字符>) |
| <标志符>:: | = <字母> (<字母> <数字>) |
| <空白>:: | = (<空格> <换行符> <制表符>) |

3.4.1.3 通信模型

智能体通信时，智能体必须知道其他智能体的名称、通信地址、能力等。可以通过两种方法实现：第一种方法让所有智能体都保存其他所有智能体的信息，这种方法浪费存储空间且不利于智能体位置的动态更新；第二种方法设置一个专门的通信服务器用来保存所有智能体的通信信息，当智能体地址改变时，由该智能体通知通信服务器，通信服务器位置不改变，所有智能体只需保存通信服务器地址^[9]。

3.4.1.4 通信服务器

通信服务器是一类特殊的智能体，在智能体的协作中起着重要的作用。在多智能体系统中，无论是一般智能体还是通信服务器都有通信部分，将它抽取出来，称为通信模块，它包含以下部分：

(1) 协议接口 为通信模块的其他部分提供一种通信方式，使它们不需要考虑通信协议。此外，当通信协议改变时，只需修改本部分，其余部分可以直接使用。协议接口包含以下内容：

- 1) 服务线程：常驻服务线程，不停监听端口地址，一旦发现有消息来，启动一个消息线程处理该消息，然后继续监听。
- 2) 消息线程：由服务线程启动，读入消息，放入接收缓冲区中。
- 3) 客户线程：由发送线程启动，将消息发出。

- 4) 计时器：超时判断。收发消息超时情况下，结束相应的消息线程或客户线程。
- (2) 接收缓冲区 缓存外界发来的消息。
 - (3) 发送缓冲区 缓存发送消息。
 - (4) 发送线程 常驻线程。不断监视发送缓冲区，发送缓冲区中一旦有消息就启动协议接口中的方法发送消息。
 - (5) 接收线程 常驻线程。不断监视接收缓冲区，一旦有消息，则进行解释和处理。
 - (6) 解释器 解释并处理通信原语。
 - (7) 地址簿 智能体的地址簿，一般智能体保留交互频繁的智能体地址；通信服务器保留已登记的所有智能体的地址。
 - (8) 消息发送函数 首先在地址簿中查询目标智能体的地址。如果没有则向通信服务器查询。得到目标智能体地址后，将消息放入发送缓冲区中。

3.4.1.5 通信服务

通信服务器提供的服务有：名字服务、查询服务、订购服务、智能体生存期服务。

(1) 名字服务 智能体将要访问的智能体的名字交给通信服务器，通信服务器查询智能体列表，返回被查询智能体的地址。在多通信服务器模型中，如果一个通信服务器不知道某个智能体的地址，它会询问其他通信服务器。

(2) 查询服务 智能体往往需要其他智能体提供某种服务。这时智能体询问通信服务器，通信服务器搜索智能体及其服务列表，找到满足条件的智能体后，返回智能体的名字和地址。

(3) 订购服务 如果系统中没有智能体需要的服务，通信服务器返回查询失败的信息，智能体可以向通信服务器申请订购服务。当有新的智能体登记系统后，通信服务器查询新增服务，当订购服务与新增服务相匹配时，通报申请订购服务的智能体。

(4) 智能体生存期服务 控制智能体的生成和终止。

3.4.2 多智能体机器人系统的数据通信与协调控制

多智能体系统的重要特征是智能体间的协调与协作^[7]。协调是指一组智能体完成一些集体活动时相互作用的性质，是对环境的适应。在这个环境中存在多个智能体并且都在执行某个动作。协调一般是改变智能体的意图，协调的原因是由于其他智能体意图存在^[8]。协作是非对抗的智能体之间保持行为协调的一个特例。当单个智能体无法独立完成目标时，需要其他智能体的帮助，这时就需要协作。协作通常建立在协调的基础上，在复杂的环境下，完成不同目标的智能体必须对其目标、资源进行协调。

协调与协作的实现都离不开通信^[10]。本节研究和实现了机器人通信系统的物理传输的底层问题，并讨论采用多智能体技术设计和实现多个类人机器人信息调度与通信系统的软件，提供良好的通信平台以实现总体控制功能。

在多机器人环境下，机器人间的协作包括两部分：机器人的任务分配和机器人内部的子目标规划。任务分配将全局目标分解并分配给系统内的一组机器人来执行。为了实现机器人间以及机器人内部功能 Agent 的实时协作，需要构建与控制体系结构相适应的协调控制模型见图 3-5，并采用合适的实时任务调度算法^[11]。本节介绍协作的底层任务分配和调度，不涉及高层的协作策略。

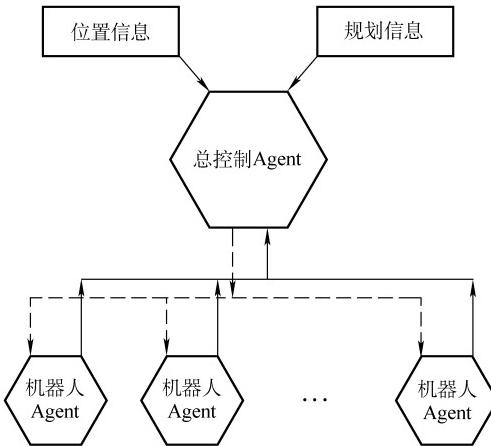


图 3-5 多机器人系统的通信系统网络结构模型

多智能体系统的重要特征是智能体间的协调与协作。协调是指一组智能体完成一些集体活动时相互作用的性质，是对环境的适应。在这个环境中存在多个智能体并且都在执行某个动作。协调一般是改变智能体的意图，协调的原因是由于其他智能体意图存在。协作是非对抗的智能体之间保持行为协调的一个特例。当单个智能体无法独立完成目标时，需要其他智能体的帮助，这时就需要协作。协作通常建立在协调的基础上，在复杂的环境下，完成不同目标的智能体必须对其目标、资源进行协调。

多智能体技术实现多个类人机器人信息调度与通信，提供良好的通信平台以实现总体控制功能。

3.5 小结

本章讲述了双足机器人技术是多种科学技术的集合体，双足机器人技术的发展是离不开这些支撑技术的，与行为数据库技术、虚拟现实技术、建模技术、多智能体通信技术密不可分。

参 考 文 献

- [1] 邓欣. 多机器人控制体系结构研究与实现 [D]. 南京: 南京理工大学, 2004.
- [2] 范永. 多机器人协作与控制研究 [D]. 北京: 中科院自动化所, 2000.
- [3] 王越超. 多机器人协作系统研究 [D]. 哈尔滨: 哈尔滨工业大学, 1999.
- [4] 杨旦, 赵春霞. 基于多智能体的微小型机器人体系结构研究与实现 [D]. 南京: 南京理工大学, 2005: 19-42.
- [5] 汤超. 双足步行机器人多电机控制及其稳定性研究 [D]. 北京: 北京科技大学, 2005.

- [6] 周远清, 张再兴. 智能机器人系统 [M]. 北京: 清华大学出版社, 1989.
- [7] 李智军, 周晓, 吕恬生. 基于群体协作的分布式多机器人通信系统的设计与实现 [J]. 机器人, 2000, 22 (4): 300-304.
- [8] 滕少冬. 应用于个人机器人的人工心理模型的研究 [D]. 北京: 北京科技大学, 2006.
- [9] 王斌, 张尧学, 陈松乔. Agent 通信模型 [J]. 中南工业大学学报, 2002, 33 (2): 209-213.
- [10] 胡志刚, 龙海涛, 钟掘. 多智能体系统中信使功能的设计与实现 [J]. 计算机工程, 2003, 29 (2): 179.
- [11] 梁峰. 多机器人系统的控制和数据通信系统的研究和实现 [D]. 北京: 北京科技大学, 2006.

第4章 双足步行机器人实验平台

步行是大多数动物所具有的移动方式，也是生物界步行方式中自动化程度最高、最为复杂的动作，它的完美实现必然要求机器人在结构设计方面产生巨大的变革和创新，从而有力地推动相关学科的发展。人类一直梦想着创造出与其构造相似、能和人类合作的拟人型机器人。而双足步行机器人是工程上少有的高阶、非线性、非完整约束、强耦合性的多自由度系统，这为机器人运动学、动力学以及控制理论的研究提供了一个理想的实验平台。

机器人步行的形式主要有双足步行、四足步行和六足步行^[1]。其中双足步行是步行方式中自动化程度最高、最为复杂的系统，同时双足步行机器人也有着其他机器人不可比拟的优点：具有适应各种地面和较强的逾越障碍的能力；步行机器人的功耗较小，通常低于轮式和履带式机器人。随着对双足步行机器人研究的不断深入，无论是影视、科幻作品还是人们对机器人的第一意识，都把像人一样的机器人作为机器人研究的最高境界。机器人的研究者也一直把实现机器人的拟人行为作为梦寐以求的目标。拟人机器人是一种智能的、机动的、能满足用户多种需求的新型机器人，这种机器人具有人的外形和基本功能，易与人共处，适应多变的活动环境。20世纪90年代前后，双足步行机器人从一般性的拟人腿部行走上升到全方位的拟人机器人研究。拟人机器人除了腿部的行走功能外，还包括手、腰和头的功能，自由度比双足步行机器人成倍地增加，与此同时也带来了控制规划、动力学、运动学上更为复杂的问题。此外，还有CCD图像处理、语音处理以及一系列传感信号的处理。拟人机器人相对于双足步行机器人的研究，更为类似人类。拟人机器人是由仿生学、机械工程学和控制工程学等多学科相互融合而成的综合性学科^[4]。

双足步行机器人的发展也为人类假肢的发展提供了有力的理论和技术支持，同时也为服务、娱乐机器人的发展开辟了新的领域^[2,5]。随着机器人的工作环境和工作任务的复杂化，双足步行机器人因其体积相对较小，对非结构性环境具有适应性较好、避障能力强、功耗低、移动盲区很小等优良的移动品质格外引人注目。拟人机器人不仅具有双腿、双臂、头、眼、颈、腰等物理特征，还能模仿人类的视觉、触觉、语言，甚至情感等功能^[3]。这和能在特种环境下工作的服务机器人是有区别的。图4-1b为北京科技大学所研制开发的双足步行机器人实验平台，该平台由双足步行机器人、PIC单片机调试器、传感器实验板、无线/有线发射模块等几部分组成。通过前面章节的介绍和分析，我们已经掌握了双足机器人的基本理论、方法和相关的支撑技术，下面就根据这些知识点，结合本书所设计的实验平台，进行有针对性的实际环节的学习。



图 4-1 双足步行机器人平台
a) 双足步行机器人外形 b) 双足步行机器人实验平台

4.1 双足步行机器人控制系统

小型双足步行机器人由于体积和电源（电池）的供电能力有限，因此对电路的结构、元器件的布线和布局都有较高的要求。对于这种要求，本书给出了模块化的系统构成，各个功能模块之间相对独立。同时，模块间通过接口的衔接来组成完整的系统。控制系统的总体框图如图 4-2 所示。

控制系统采用多级分布控制方法，分为上、下两层控制，其功能分别为：

(1) 上层 PC，完成机器人行为数据的离线计算和调整，通过有线或者无线网络控制和协调各个机器人的动作行为，并且可以通过机器人所带的传感器获取环境信息。

(2) 下层 实现控制算法，控制多路舵机协调运动，其控制系统可以分为以下几个模块：

1) 电源模块：向下位机系统各模块提供稳定的电压。

2) 数据存储模块：由 I²C 总线的串行 E²PROM (24C512) 组成，PIC16F877 通过 I²C 总线对其进行读写操作。

3) 有线通信模块：由 SP3232 电平转换芯片实现，通过 RS232 串行总线实现上、下位机的数据交互。

4) 无线通信模块：由 nRF905 射频芯片和 STC2051 微控制器组成，通过射频 (RF) 实现上、下位机的数据交互。

5) 电动机驱动模块：由 PIC16F877 主微控制器组成，实现控制算法、下位机协调和控制多路舵机控制信号的生成。

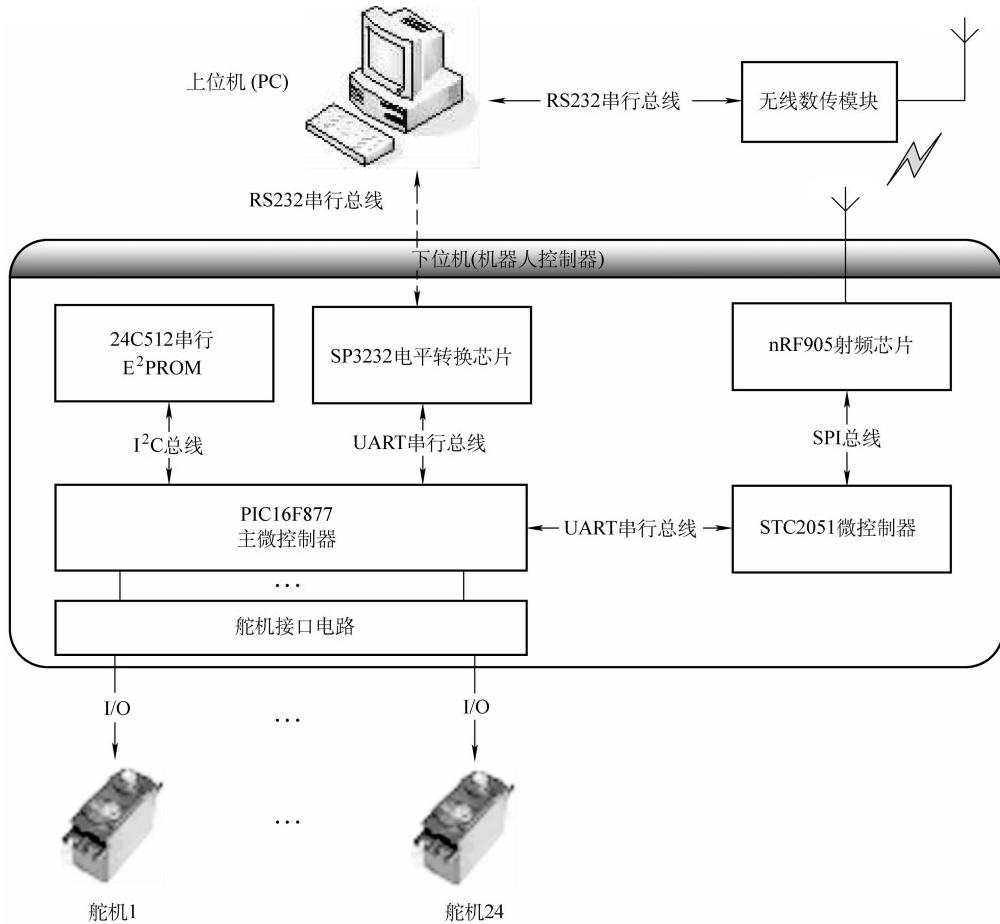


图 4-2 控制系统的总体结构框图

4.2 双足步行机器人的上位机软件

4.2.1 初始化设定模块

在机器人的安装过程中，由于关节的可移动性，使得组装后的机器人都有着各自不同的初始状态。在这种情况下，如果我们仍然采用同一参数对机器人进行初始化，则最终会发现，所有的机器人姿势各不相同，而原因就在于组装时的关节误差。因此我们提出了初始位置（Homeposition）的概念。

初始位置是机器人良好直立、腿部各舵机受力均匀时动作的名称。我们所调试完成的每一个动作数据（如上所述）的开始和最后的动作都是使用的这个动作。初始位置是在每个机器人安装完毕后，首先调试出的。通过对不同机器人的初始位置进行比较，我们不难发现它们基本上都有所不同，尽管这其中应考虑调解设定时的人为因素，但同时也说明每台舵机

实现相同的角度都是不尽相同的，如图 4-3 所示为标准 90° 和显示 90° 实际带偏差舵机。

对机器人的操作中，初始化的设定有着非常重要的意义。对于我们来讲，不可能针对每一台机器人都重复同样的工作去调试数据，这样做不但浪费时间，而且未能从根本上解决问题的所在。

因此，如果不作任何处理，机器人动作数据的可移植性非常差，即编号为 Z1 的机器人某组已经调试成功的动作，直接应用在编号为 Z2 的机器人上效果可能非常不好，基本上需要重新做大量的调试工作。另外，如果某台机器人身上的某一舵机损坏更换，则其原来的某些动作数据也会因为舵机的更换而变得不能使用。

为此，我们仍应用初始位置来解决。尽管由于工艺问题，每台舵机对于同一角度的位置并不相同，但是我们可以人为地进行校正，即把初始位置作为一个参考基准，而今后的所有动作只需要在此舵机角度上进行相应的叠加就可以完成，我们把与其进行叠加的动作数据称为给定控制角，也称相对数据，而下位机实际收到的数据是叠加后的结果，即实际控制角，也称绝对数据。三者之间关系如下式所示。

$$\text{实际位置角度}(\circ) = \text{相对转动角度}(\circ) + \text{初始位置(Homeposition)角度}(\circ)$$

(4-1)

这样，我们无须因为舵机工艺的原因对于每台机器人的动作数据进行调试，而只需要调试出一台机器人的实际控制角动作数据，根据式(4-1)求出相对动作数据，即给定控制角，进而保存，就可以反向运用式(4-1)得到新的机器人动作实际控制角数据，只是需要注意，对于不同的机器人，使用公式时，初始位置项均应为对应机器人的实际值。关于设定初始位置的操作如图 4-4 所示。



图 4-4 初始设置界面

因此我们需要设定这样一个初始数据，使得机器人的相对初始位置都是完全一样的。那么，同样一组数据，对于所有的机器人都可以使用，并做出同样的动作。这样不仅可以免去对每一台机器人都重复做同样的调试数据工作，节约了时间。另一方面，只要我们在初始位置的调试上对每台机器人都进行严格的规范，就可以同时保存多台机器人的初始位置，而只保存一套动作数据，通过动作数据与初始位置的简单计算，为多个机器人提供统一的动作数据，使机器人的动作数据具有通用性，从而达到动作标准统一。这是我们进行多机器人表演

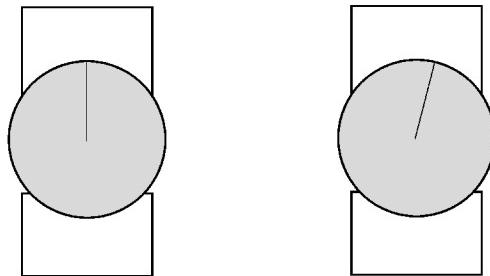


图 4-3 标准 90° 和显示为 90° 实际带偏差舵机

时动作质量的保证。初始位置、相对动作、机器人动作数据以及与音乐、节拍、情绪状态的关系如图 4-5 所示。

4.2.2 机器人的运行模式与通信协议

在通信中，需要上位机和下位机进行数据通信，而异步串行通信是一种常用的通信手段。通信协议也叫通信规程，是指通信双方格式上的约定。数据通信中，在收/发信器之间传送的是一组二进制位串，但它们在不同的位置可能有不同的含义，有的只是用于同步，有的代表通信双方的地址，有的是一些控制信息，有的则是通信中真正要传输的数据，还有的是为了差错控制而附加上去的冗余位。这些都需要在通信协议中事先约定好，以形成一种收/发双方共同遵守的格式。在逐位传送的串行通信中，接收端必须能识别每个二进制位从什么时候开始，这就是位定时。通信中一般以若干位表示一个字符，除了位定时外，还需要在接收端能识别每个字符从哪里开始，这就是字符定时。

异步串行通信时，每个字符作为一个独立的信息，可以随机出现在数据流中，即每个字符出现在数据流中相对时间是任意的。然而，一个字符一旦开始出现，字符中各位便以预先固定的时钟进行传送。因此异步通信方式的“异步”主要体现在字符与字符之间，而同一字符内部的位与位间是同步的。为确保异步通信的正确性，必须找到一种方法，使收发方在随机传送的字符内部实现同步。这种方法就是在字符格式中设置起始位和停止位，即在一个字符正式发送之前先发一个起始位，该字符结束时再发一个停止位。接收器检测到起始位便知道字符到达并开始接收字符，检测到停止位则知道字符传输已结束。由于这种通信协议是靠起始位和停止位来实现字符内部同步的，所以有时也称为起止式协议。

下位机与上位机 PC 之间的通信，首先规定好彼此之间的通信协议，才可以实现上位机软件与下位机软件的握手。软件握手最大的弊端是传输速度慢，因为每发送一帧或一个字节的数据都必须实现一次握手过程，这就影响到了传输的速度。选择软件握手是基于：软件握手接收数据的灵活性提高了。采用软件握手，其协议完全是由软件给出，因此可以灵活地加以改变，以适应不同的场合。

在进行数据通信的软件设计时必须解决好两个方面的问题：一是可靠性，二是速度，而可靠性是主要的。速度只能是在可靠的基础上实现。可靠快速传输的实现需要上下位机软件以及通信协议等各个环节及其间相互配合。完成硬件层的串行通信设计较为简单，然而在实际中，干扰是不可避免的，因而仅仅依靠硬件层的通信必然会出现问题。为了实现可靠通信，就必须应用较为可靠的基于帧的通信协议进行各种数据的传输。

对于串行通信而言，程序的可靠性是由协议保证的。通信的可靠性主要体现在所使用的通信协议上，主要有以下几个方面：

- (1) 通信之前先握手 通过握手的方式可以判断硬件是否正常，计算机是否处于等待

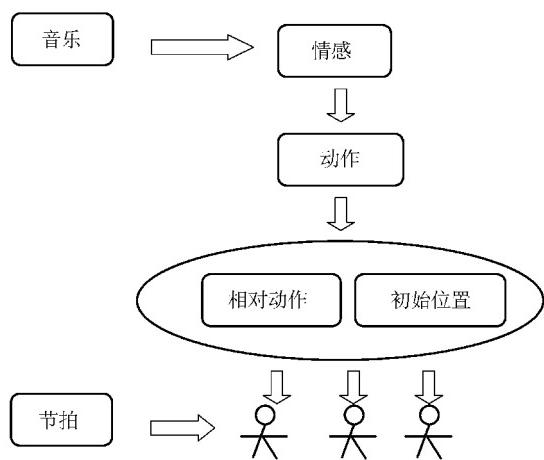


图 4-5 初始位置与机器人相对动作数据的关系

状态。每次握手之间等待一段时间，以确保计算机状态。

(2) 数据传输使用帧的方式 这样整帧数据不可能全部丢失。校验字将整帧信号进行校验，使误收的可能性很小。

(3) 帧方式通过判断帧头起始字符来决定一帧的开始 这样就避免了部分数据进入到内部数据处理之中，这使数据处理错误的可能性减少，并可防止其他串行数据和要求数据混淆。

(4) 丢包一次也不影响程序运行 即使在强干扰环境下，对于单片机接收而言，也只是数据传输错误。

在机器人的舞蹈动作表演中，上位机要向机器人发送很多指令和参数，如：机器人的运行模式、动作参数、处理出错重传和多个机器人之间的动作配合等。同样的，机器人也有很多重要反馈参数，如机器人初始位置、完成动作、动作数据/指令发送错误等信息。制定合理的通信协议，使机器人的表演更具灵活性和交互性，以达到机器人表演的流畅，并且使机器人之间协调配合。机器人端的工作流程如图 4-6 所示。

通信协议可分为两种模式：

(1) 同步模式 在此模式下，控制端实时地向机器人发送数据，机器人可以实时地根据指令进行表演，具有很好的灵活性和交互性。但是由于控制端与机器人的实时通信过程中没有很好的可靠性，并且不能够全面地掌握机器人的状态，所以动作的安全和质量就不容易得到很好的保证。

(2) 发送模式 在此模式下，控制端提前向机器人发送数据存储在机器人的存储器中。执行动作时，机器人按照预先设定好的指令进行表演，具有很好的可靠性，并且保证机器人的动作质量。但是由于控制端与机器人没有很好的实时通信过程，所以灵活性和交互性就不能得到很好的保证。

控制端的工作流程如图 4-7 所示。

4.2.3 双足步行机器人控制端软件

上位机的控制端软件是双足步行机器人平台的重要组成部分，在整个系统中，为嵌入式系统发挥着后台支撑作用。PC 的功能很强大，不管是硬件资源还是软件资源，嵌入式系统都无法和 PC 相比。有了这一强大的后盾，我们对机器人的操作将变得更加形象、便捷和丰富多彩。上位机软件由主界面、动作调试模块、基于音乐情绪的列表界面及情感动作数据库^[6]、动作数据库、三维仿真模块和通信模块组成。软件的功能模块如图 4-8 所示。

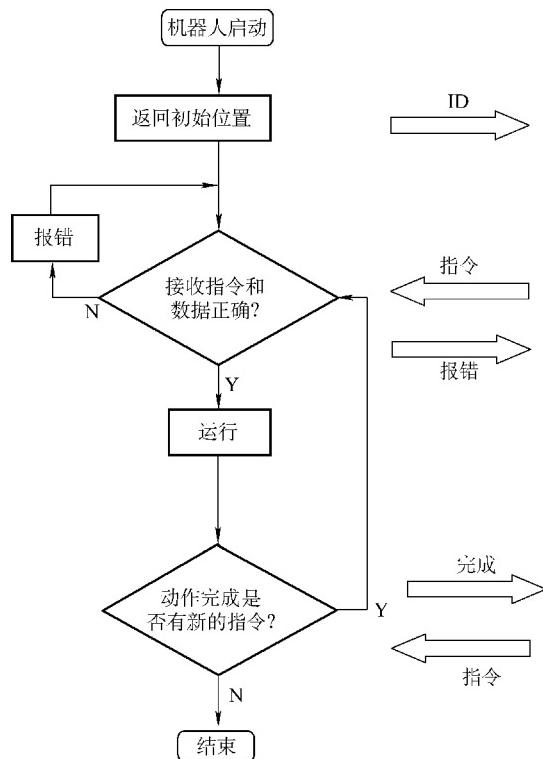


图 4-6 机器人端的工作流程及数据交互

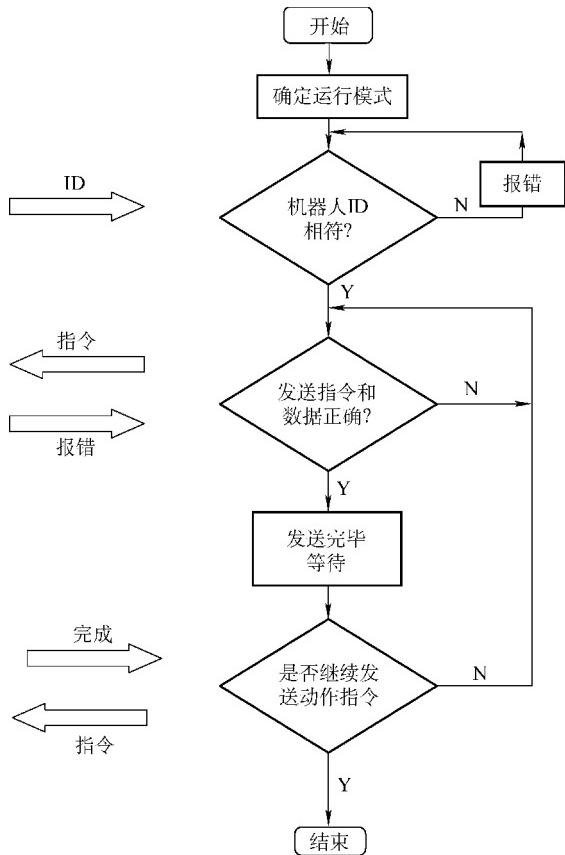


图 4-7 控制端的工作流程及数据交互

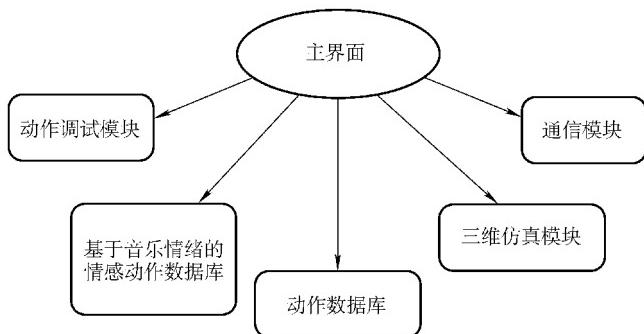


图 4-8 上位机软件主要功能模块

下面，就对上位机软件平台的主要功能模块进行说明。

(1) 主界面、动作调试及动作数据库 上位机主界面主要功能是进行机器人的初始位置以及动作调试；机器人端与 PC 控制端的串行口通信的开关及数据的监视；调试后的动作可以以 .dat 文件的形式保存实际控制角动作数据或者以相对动作数据保存在动作数据库中，如图 4-9 所示。



图 4-9 上位机软件主界面

(2) 音乐列表界面及情感动作数据库 通过该界面可以播放音乐，管理基于情绪模型的音乐数据库^[8]，并可以根据音乐的音频快慢在情感动作数据库中查找到双足步行机器人的
情绪参数，在动作数据库中查找与情绪系数最接近的动作组，自动地向机器人发送与所播
放音乐情感特征相符的动作数据^[9,10]，如图 4-10 所示。根据操作者想表达的音乐情绪在数
据库中显示与所选情绪最为接近的动作数据进行动作的编排^[11]，并且保存动作组。同时可
以根据所选的机器人 ID 号查找数据库中的机器人初始位置，从而生成针对该机器人的动作
数据组^[12,13]，如图 4-11 所示。

(3) 三维仿真 根据双足步行机器人的实际结构建立的三维立体模型，可以在虚拟的
环境中为双足步行机器人创作各种动作，如街舞、倒立等，再将得到的动作数据下载到双足
步行机器人上，从而实现机器人的动作。

(4) 通信 通过 RS-232 串行口控制双足步行机器人的动作演示，可以在线地编排双足步
行机器人的动作，给双足步行机器人发动作数据，发送数据的同时，在串口监控模块窗口中显
示发送错误的动作帧见图 4-9，并重新发送该动作帧，以确保准确无误地发送动作数据。

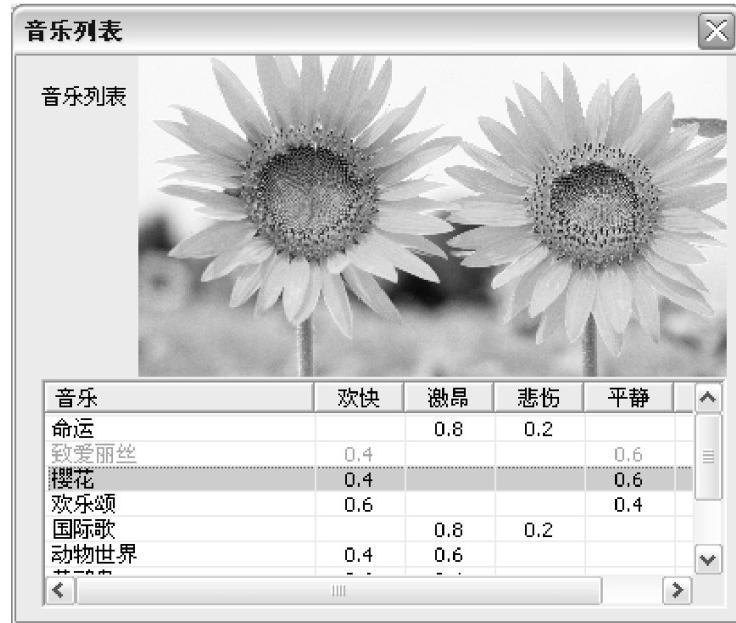


图 4-10 上位机软件音乐情感界面



图 4-11 上位机软件动作编排界面

4.2.4 多机器人控制

多机器人系统的通信是研究多机器人系统的基础，由个体机器人组成一个群体系统，通信是必不可少的。多机器人系统在执行某项任务时，为了实现协调与合作，个体机器人的传感器必须提供足够的环境描述信息和其他机器人的信息，因此机器人个体之间或者上层控制和下层合作之间的通信是必要的^[7]。机器人之间的通信方式主要有两种，即直接通信和间接通信。直接通信要求发送和接收信息能保持一致性，因此机器人之间需要一种通信协议，

而且直接通信时发送方和接收方必须同时在线，间接通信没有此项要求。一般来说，直接通信存在于有智能的机器人之间，而间接通信存在的范围就比较大，如个体和个体通信、个体和群体通信、个体和环境通信等。目前，大部分关于多机器人的通信主要采用广播的方式，即个体机器人将自己的位置和传感器信息以及自己从事的工作信息广播出去，其他个体机器人可以按自己的需要选择信息，或主控机器人通过广播分配任务等。

与分层体系结构相对应，多机器人系统的通信模型也采用分层结构。为了便于集中管理，客户机/服务器模型示意图如图 4-12 所示。

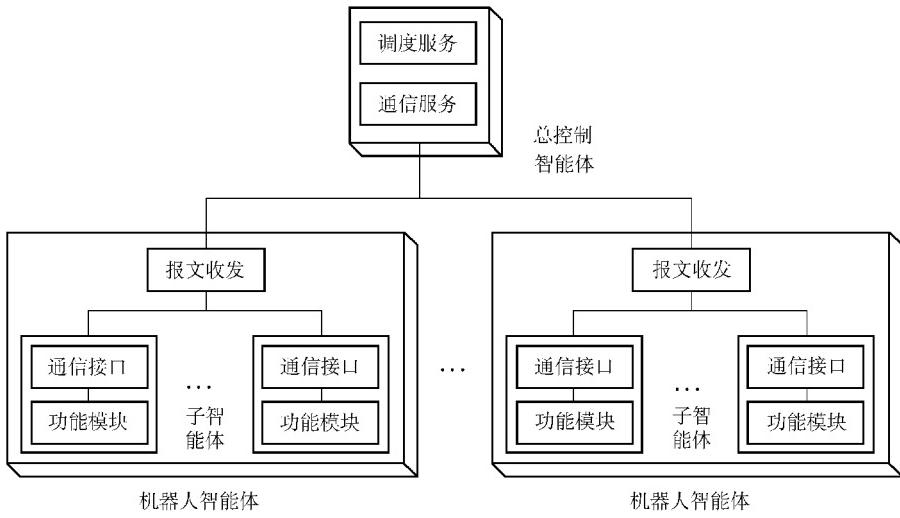


图 4-12 分层通信模型示意图

在图 4-12 中，通信模型总体上采用两层客户机/服务器 C/S 结构。控制中心与各机器人的通信服务器构成第一层，机器人通信服务器与功能模块构成第二层。控制中心的通信服务器负责与各机器人的通信服务器交互，实现信息的传递。当收到机器人通信服务器的连接请求时，总控制中心通信服务器开启线程为之服务，同时为其分配端口号，将机器人名称和端口号存入地址表中备查；机器人通信服务器是控制中心的客户端，而对功能模块而言，它是服务器，负责各功能模块间的数据交互，实现分布式计算；通信接口仅为客户端，负责指令的生成和解释，功能模块通过通信接口发送和接收数据。

为了提高系统适应环境的能力，整个系统动态可扩充。关于模型的动态可扩充能力，我们可以将其分成两层理解，一是机器人的动态接入，由于采用了便于管理的 C/S 结构模型，在系统运行过程中需要接入机器人系统时，该机器人的调度进程向控制中心进程发出接入请求，控制中心进程启动新线程为之服务，当机器人系统退出时，控制中心进程结束相应的服务线程，释放资源；与此类似，在机器人系统内部，由于机器人的状态改变，需要启动或停止部分工作进程（线程）。启动工作进程时，该进程所在机器人的调度进程向工作进程发出启动命令；停止工作进程时，则发出命令将该工作进程挂起。工作进程如图 4-13 所示。

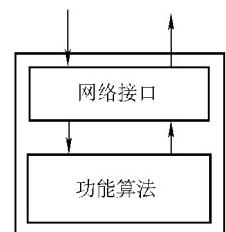


图 4-13 工作进程结构

每个工作进程都包含两部分：网络接口和功能算法实现，算法实现部分并不需要考虑通

信问题，它仅关注功能的算法实现，通过定义接口规范，可以保证功能算法部分的较强封装性，从而使算法的可移植性等功能大幅度加强。

4.3 总体设计架构

我们希望双足机器人平台可以进行基于传感器的特征提取，对各种环境信息和用户意图及情感信息获取的多模式行为关联融合，从而实现个性化交互的双足机器人控制端软件的总体设计架构，如图 4-14 所示。

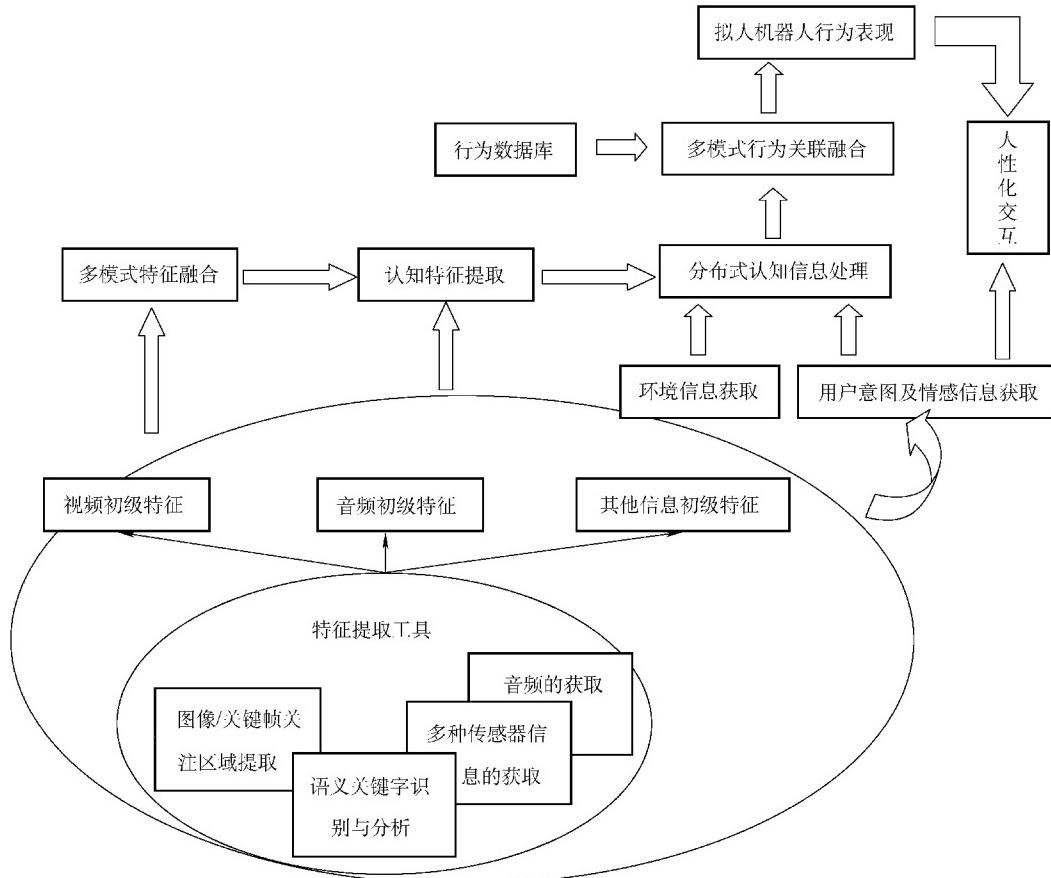


图 4-14 双足步行机器人的总体设计架构图

4.4 小结

本章介绍了用于音乐表演研究的双足步行机器人试验平台，将其控制系统分为上、下两层控制。上层为控制端软件的软件架构，由动作调试模块、音乐情绪数据库、动作数据库、通信模块、动作编排模块组成；下层为整体硬件架构，包括控制电路、多机器人通信协议与控制端通信的各种控制运行模式。提出初始位置的概念用于软件平台的初始化设定模块，以克服机器人关节有各自不同的初始位置的问题，使机器人的动作数据具有通用性。这是我们

进行多机器人标准化动作表演的基础。最后，提出了基于传感器特征提取各种环境信息及情感信息获取的多模式行为关联融合，从而实现个性化交互的双足机器人平台的总体设计架构。

参 考 文 献

- [1] “机器人”王国将建立-机器人是大产业 [OL]. 2007, 03. <http://tech.QQ.com>.
- [2] 阮晓钢, 仇忠臣, 关佳亮. 双足行走机器人发展现状及展望 [J]. 机械工程师, 2007 (2).
- [3] 李允明. 国外仿人机器人发展概况 [J]. 机器人, 2005, 27 (6).
- [4] 满翠华, 范迅, 张华, 等. 类人机器人研究现状和展望 [J]. 农业机械学报, 2006, 37 (9).
- [5] 王伟. 从机器人热潮到机器人产业化的发展 [J]. 机器人技术与应用, 2006 (4): 10-14.
- [6] 黄虹, 蔡黎曼. 音乐的熟悉性-复杂性-情感类别与偏好的关系研究 [J]. 中国音乐学, 2007 (2): 131-140.
- [7] 车玲玲. 基于双足机器人平台的控制策略与仿真研究 [D]. 北京: 北京科技大学, 2006.
- [8] Nagashima K, Kawashima J. Experimental study on arranging music by chaotic neural network [J]. International Journal of Intelligent Systems, 1997 (12): 323-339.
- [9] Margaret I, Johnson. Toward an expert system for expressive musical performance [J]. Computer, 1991: 24 (7):30-34.
- [10] Katayose H Fukuoka T, Takamietal K. Expression extraction in virtual music performances [J]. Pattern Recognition, 1990: 780-784.
- [11] 刘丹, 张乃尧, 朱汉城. 音乐特征识别的研究综述 [J]. 计算机工程与应用, 2002 (24).
- [12] 白晨. 用于机器人表演模式的音乐情感计算方法与技术研究 [D]. 北京: 北京科技大学, 2007.
- [13] 王国江, 人机情感交互的方法与技术的研究 [D]. 北京: 北京科技大学, 2006.

第5章 双足步行机器人的行为生成技术

通过前面的学习，我们已经掌握了双足步行机器人的基本技术和方法，并根据实验平台进行有针对性的分析和研究，但要双足步行机器人模仿人类高级智能，就要研究双足步行机器人的行为生成技术。本章重点讲述拟人情感建模和拟人行为生成。

目前人工心理与情感计算应用研究是拟人机器人技术的重要内容之一^[6,7]。如何将情感转移的控制思想与机器人具体控制技术相结合是高科技技术的象征^[1,4]。本章将从双足步行机器人的多智能体控制的实现、情感模型的搭建^[8]、行为决策的设计、情绪熵的选取、语音数据库的设计与机器人动作等方面进行详细描述。这是基于第4章介绍的双足步行机器人实验平台来展开的。该机器人共有17个自由度。通过计算机离线计算机器人的行为数据，进而协调17个舵机的运动，从而实现对机器人的控制。

5.1 多智能体控制的实现

5.1.1 多智能体的特点

随着计算机技术、超大规模集成电路、控制理论、人工智能理论、传感器技术等的不断成熟和发展，由多学科交叉而形成的机器人学研究也进入了一个崭新的阶段。从可编程的、示教再现型的工业机器人到具有一定传感能力、一定适应能力的机器人，再到配备多种先进传感器、具有较强的适应能力的智能机器人，机器人学的研究工作经历了一个从简单到复杂，从功能单一到功能多样，从工业制造领域到军事侦察、核工业、航空航天、服务业、医疗器械、基因工程等诸多领域的过程。可以预见，在不久的将来，机器人技术在各个领域的应用将会更加广泛和深入。而各种机器人系统在实际工作中的广泛应用又为机器人学提出了新的要求和新的研究课题。多机器人系统的研究就是在这些新的应用需求驱动下提出的，并随着机器人学的不断发展而逐渐成为机器人学研究的一个重要分支。

在机器人研究的早期，单机器人的结构、运动学、控制和信息处理是研究的重点。随着机器人技术的发展，单个机器人的能力、鲁棒性、可靠性、效率等都有很大的提升。但面对一些复杂的、需要高效率的、并行完成的任务时，单个机器人则难以胜任。为了解决这类问题，机器人学的研究一方面要进一步开发智能更高、能力更强、柔性更好的机器人，另一方面要在现有机器人的基础上，通过多个机器人之间的协调工作来完成复杂的任务^[3]。

多机器人系统是机器人学在新的应用需求驱动下提出的、具有重要理论价值和应用前景的研究方向^[9]。从系统的角度出发，对多个机器人组成的群体进行深入研究，充分发挥多机器人系统所具有的系统冗余、并行工作、资源分布等优势。

多机器人系统的研究是从单个机器人系统的研究扩展开来的，但区别于单个机器人系统，多机器人系统的特点可以概括如下：

- (1) 空间分布 多个机器人可以在工作空间的不同区域同时工作^[11]。

- (2) 功能分布 功能不同的机器人或具有不同任务（目标）的多机器人可以协同工作。
- (3) 时间分布 多个机器人可以执行时间分布的任务。
- (4) 信息分布 多个机器人可以具备相同的知识或不同的知识，通过通信，协作机器人可以进行知识的交换和学习^[10]。
- (5) 资源分布 多机器人系统中各机器人可以具有不同的传感器和执行器。

5.1.2 多机器人动作协调设计及转化程序的实现

双足步行机器人的动作主要是通过 17 个舵机共同协调转动来得以实现的。每个动作包含 1 个速度位和 17 个舵机的角度值，算作一帧动作。一个基本动作包含若干帧动作，这些帧也就是字符都存放在相应数组中，通过上位机将数据完全发送到下位机，如图 5-1 所示。

| 序号 | DATA | SPEED | CH1 | CH2 | CH3 | CH4 | CH5 | CH6 | CH7 | CH8 | CH9 | CH10 |
|----|------|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 1 | DATA | 6 | 166 | 160 | 94 | 78 | 5 | 6 | 82 | 80 | 57 | 58 |
| 2 | DATA | 6 | 161 | 141 | 94 | 78 | 2 | 21 | 82 | 78 | 110 | 126 |
| 3 | DATA | 3 | 161 | 136 | 84 | 99 | 2 | 21 | 82 | 95 | 110 | 125 |
| 4 | DATA | 6 | 161 | 136 | 84 | 99 | 2 | 21 | 82 | 95 | 107 | 145 |
| 5 | DATA | 6 | 161 | 136 | 84 | 99 | 114 | 27 | 82 | 95 | 79 | 149 |
| 6 | DATA | 6 | 161 | 136 | 84 | 99 | 2 | 21 | 82 | 96 | 157 | 116 |
| 7 | DATA | 6 | 161 | 136 | 84 | 99 | 2 | 21 | 82 | 96 | 157 | 116 |

当前状态
数据读取完成
传送进度
发送

运行
复位
顺序
确定
退出

图 5-1 上位机控制平台显示动作数据

对于实验所运用到的双足步行机器人的基本动作可以分解为如图 5-2 所示动作。

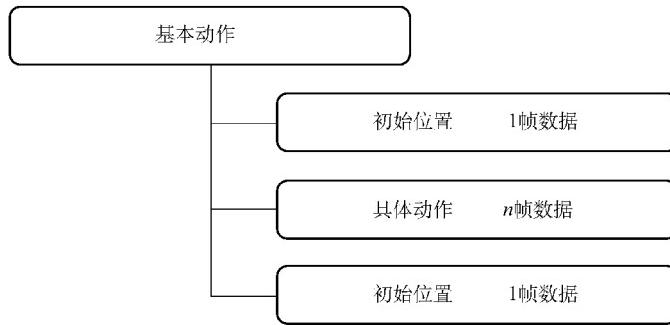


图 5-2 机器人基本动作的分解

初始动作可以作为双足步行机器人的复位动作，也就是任何基本动作的开始和收尾都是以此动作为结尾。这里我们选择初始位置（Homeposition）为双足步行机器人竖直站立的姿态，如图 5-3 所示。因此，双足步行机器人的基本动作设计主要放在了除了初始位置的其他帧中。

复杂动作莫过于双足步行机器人的行走，目前世界上大多数双足步行机器人系统都采用 ZMP 作为稳定行走的判据。当双足机构处于动态平衡时，ZMP 和脚底所受地面反力的压力中心（Center of Pressure, CoP）是重合的。因此可以根据检测到的地面反力信息，计算 CoP，

通过控制策略调整 ZMP 和 CoP 的位置，使两者重合，实现机器人的动态稳定行走。行走模型如图 5-4 所示。

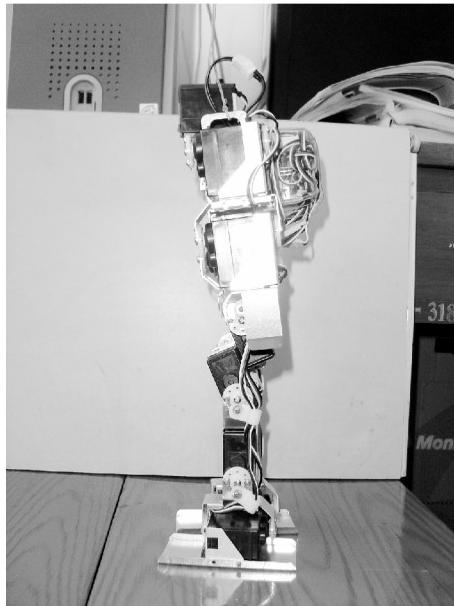


图 5-3 机器人初始位置

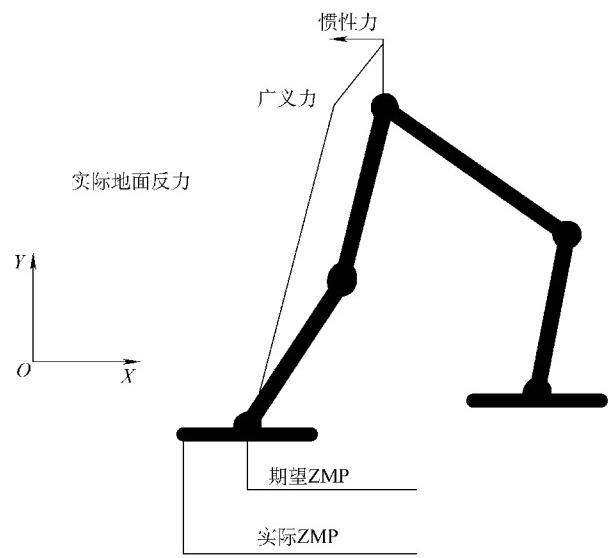


图 5-4 ZMP 行走模型示意图

根据力学原理可知，当物体处于静止状态时，其平衡的充要条件是其重心在地面上的投影落在其支撑面内，而广义地讲，当物体处于运动状态时，其平衡的必要条件是所受重力与惯性力的合力的延长线通过其支撑面内（该合力的延长线与支撑面的交点称为 ZMP）。

以前实验室中传统的调节机器人的动作都是用手动调节上位机控制平台的滚动条控件来进行编写的，这样调节机器人的动作，对于单个机器人来讲，优点是精确，符合精确的步态发生模型的要求。例如实现双足步行机器人的行走，为了符合 ZMP 理论要求，需要精确地拖动上位机控制平台的滚动滑块来进行调解。

然而对于多机器人的动作协调设计，则需要有更加有效率的编写机器人动作文件的方式，而不能拘泥于以前的传统滚动条的编写方法。为了使机器人的动作协调设计得以顺利实现，新的机器人动作编写方法是必需的。

在研究本实验平台的双足步行机器人的动作当中，每台机器人的动作即使表达一样，但它们所对应的数据却不尽相同，这是由于每台机器人的初始位置所对应的数据不同，这是舵机本身性能所带来的无法避免的硬伤。换句话说，用 17 个舵机组成新机器人的初始位置数据就与其他 17 个相同性能品牌的舵机组成的已有机器人的初始位置不同。

在反复的学习与研究中，从双足步行机器人每帧的动作数据之间着手分析，得出了双足步行机器人新动作转化的公式。

在此令机器人实际动作时的舵机角度值 (A_1) 与初始位置的角度值 (H_1)、机器人进行动作时舵机转动的角度值 (R_1) 三者之间的关系，得出机器人实际动作时角度值关系公式：

$$R_1 + H_1 = A_1 \quad (5-1)$$

在三个变量中，只有舵机转动的角度值 R_1 是恒定不变的。这样就可以将公式进行转化，

用以通过数学关系来寻找新机器人的动作数据与已有机器人的动作数据的关系。

令已有机器人实际动作时的舵机角度值 (A_1) 与初始位置角度值 (H_1)、机器人进行动作时舵机转动的角度值 (R_1)，那么可以将式 (5-1) 转化成

$$R_1 = A_1 - H_1 \quad (5-2)$$

再令新机器人实际动作时的舵机角度值 (A_2) 与初始位置角度值 (H_2)、机器人进行动作时舵机转动的角度值 (R_2)，这时对于新机器人来说，做相同动作时， R_1 与 R_2 相同，即

$$R_1 = R_2 \quad (5-3)$$

则得到

$$A_1 - H_1 = A_2 - H_2 \quad (5-4)$$

调试机器人的新动作就是求新机器人的设计动作时的舵机角度值 A_2 。

若已知某台双足步行机器人实际动作时的舵机角度值 (A_1) 与初始位置角度值 (H_1)、通过调试机器人的三大步骤的第一步，可以得到初始位置角度值 (H_2)，即得到动作转化公式为

$$A_2 = A_1 - H_1 + H_2 \quad (5-5)$$

这个公式就是双足步行机器人新动作转化程序的算法精髓。

在 VC++ 环境中，实现双足步行机器人新动作转化程序的主要函数如下：

```
void CNewActionDlg::OnChange()
{
    // TODO: Add your control notification handler code here
    CFile Chfile;
    unsigned char temp[65535];
    unsigned char uchData[3600][18];
    int Num_data;
    int i,j;
    if(Opentest && Savetest)
    {
        BOOL test = Chfile.Open(OpenFileName,CFile::modeRead);
        UINT nbytes = Chfile.Read(temp,65535);
        Chfile.Close();
    }
    test = Chfile.Open(SaveFileName,CFile::modeWrite|CFile::modeCreate);
    Num_data = nbytes/18; //求得文件数据总行数
    int unum = 0;
    for(i = 0; i < Num_data; i++)
    {
        unum++; //计数
        for(j = 0;j<18;j++)//给发送的数组赋值
    }
}
```

```

uchData[ i ][ j ] = temp[ 18 * i + j ] ;
}

UpdateData( true ) ;

unsigned int
Homp1[ 17 ] = { m _ Dh1 , m _ Dh2 , m _ Dh3 , m _ Dh4 , m _ Dh5 , m _ Dh6 , m _ Dh7 , m _ Dh8 , m _ Dh9 , m _ Dh10 , m _ Dh11 , m _ Dh12 , m _ Dh13 , m _ Dh14 , m _ Dh15 , m _ Dh16 , m _ Dh17 } ; //已有机器人的初始位置的 17 个数据放在数组 Homp1[ 17 ] 中
unsigned int
Hompnew[ 17 ] = { m _ Ch1 , m _ Ch2 , m _ Ch3 , m _ Ch4 , m _ Ch5 , m _ Ch6 , m _ Ch7 , m _ Ch8 , m _ Ch9 , m _ Ch10 , m _ Ch11 , m _ Ch12 , m _ Ch13 , m _ Ch14 , m _ Ch15 , m _ Ch16 , m _ Ch17 } ; //新机器人的初始位置的 17 个数据放在数组 Hompnew[ 17 ] 中
for( j = 1 ; j < 18 ; j + + )

{
    if( uchData[ i ][ j ] + Hompnew[ j - 1 ] < = Homp1[ j - 1 ] )
        uchData[ i ][ j ] = 1 ; //机器人舵机角度值不能小于 1 ,强行令 uchData[ i ][ j ] = 1
    else uchData[ i ][ j ] + = Hompnew[ j - 1 ] - Homp1[ j - 1 ] ;
    // A2 = A1 - H1 + H2 公式算法的程序中的实现
    if( uchData[ i ][ j ] > = 180 )
        uchData[ i ][ j ] = 179 ;
}
}

Chfile. Write( uchData , 18 * unum ) ;
Chfile. Close( ) ;

OpenFileName = '\0 ' ;
SaveFileName = '\0 ' ;
MessageBox( "转换成功" , "转换结束" , MB _ OK ) ;
Savetest = FALSE ; //将“存储成功标志”重新设定为 false
Opentest = FALSE ; //将“文件打开成功标志”重新设定为 false
}

由此，多机器人的新动作转化程序就实现了，程序运行界面如图 5-5 所示，大大提高新组装机器人的动作文件编写效率。这也为双足步行机器人的多智能体控制做出了支持，为机器人团队表演和“千手观音”表演做出了相应充足的准备。

```

5.1.3 “千手观音” 机器人多智能体控制的实现

通信是机器人之间进行交互和组织的基础。通过通信，多机器人系统中各机器人了解其他机器人的意图、目标和动作以及当前环境状态等信息，进而进行有效的控制，协作完成任务。一般来说，机器人之间的通信如图 5-6 所示，可以分为隐式通信和显式通信两类。

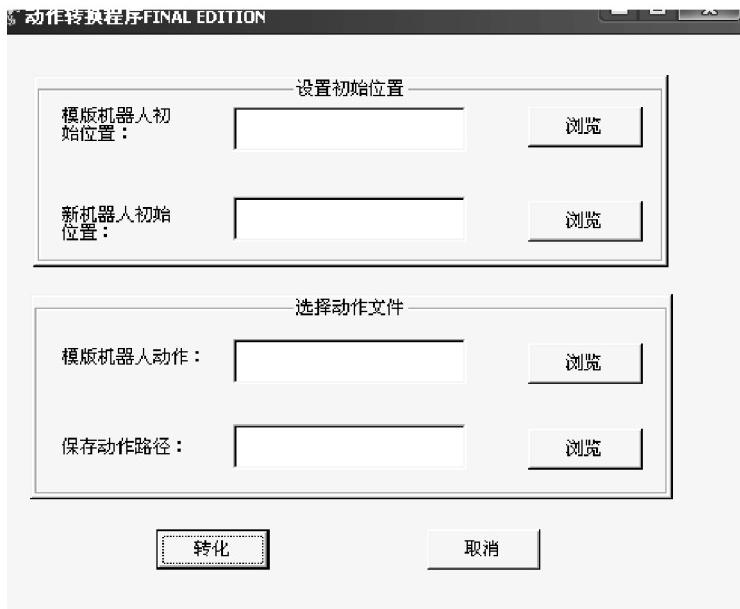


图 5-5 新动作转化程序的运行界面

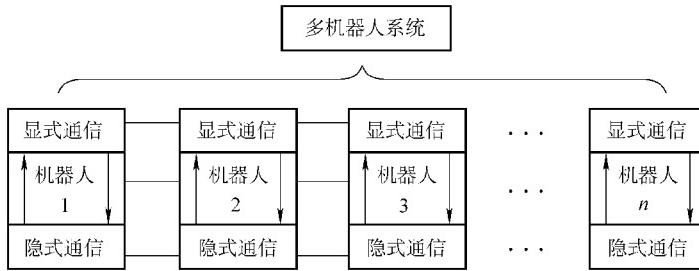


图 5-6 多机器人系统的通信

使用隐式通信的多机器人系统通过外界环境、自身传感器来获取所需的信息，并实现相互之间的协作，机器人之间没有通过某种共有的规则和方式进行数据转移和信息交换来实现特定信息的传递。在使用隐式通信的多机器人系统中，由于各机器人相互之间不存在数据、信息的显式交换，所以多机器人系统可能无法使用一些高级的协调协作策略，从而影响了其完成某些复杂任务的能力。

使用显式通信的多机器人系统利用特定的通信介质，通过某种共有的规则和方式实现特定信息的传递，因而可以快速、有效地完成各机器人间数据、信息的转移和交换，实现许多在隐式通信下无法完成的高级协调协作策略。

与已有的通信系统不同，此系统不需一个中心服务器或基站的存在，符合分布式机器人系统的基本特点和要求，并通过共享无线通信通道的方式实现了机器人之间的数据通信。Eiichi Yoshida 等人从通信容量和成本的角度对多机器人系统的局部通信进行了研究，提出了

在许多机器人随机发送消息的环境中对多机器人系统局部通信效率进行优化的方法，以传输等待时间最小为目标，通过使用信息成功传送的概率进行分析计算，推导出了各机器人最优的通信范围。多机器人系统的显式通信虽然可以强化机器人之间的协调协作关系，但也存在以下问题：各机器人的通信过程延长了系统对外界环境变化的响应时间；通信带宽的限制使机器人之间的信息传递、交换出现瓶颈；随着多机器人系统中机器人数目的增加，通信所需时间大量增加，信息传递中的瓶颈问题更为突出。

隐式通信与显式通信是多机器人系统各具特色的两种通信模式，如果将两者各自的优势结合起来，则多机器人系统就可以灵活地应对各种动态未知环境，完成许多复杂任务。利用显式通信进行少量的机器人之间的上层协作，通过隐式通信进行大量的机器人之间的底层协调，在出现隐式通信无法解决的冲突或死锁时，再利用显式通信进行少量的协调工作加以解决。这样的通信结构既可以增强系统的协调协作能力、容错能力，又可以减少通信量，避免通信中的瓶颈效应。

每个机器人其实对应的就是其中的单个智能体，通过上位机程序和无线通信模块来对多个机器人进行控制，实现了一对多的多智能体控制，发送数据的函数程序编写如下：

```
DWORD CROBCONDlg::WorkThread() //发送数据
{
    .....
    uchTemp[0] = 0xff
    uchTemp[1] = 0xfe;
    uchTemp[2] = m_RobotID; // 存放机器人公共 ID = 127 以达到同时对多智能体控制
    uchTemp[3] = 0x04;
    uchTemp[4] = m_unDataGroupNum; //存放动作组号
    uchTemp[5] = (unsigned char)i;
    uchTemp[6] = (unsigned char)unum;
    uchTemp[7] = uchData[i][0];
    for( int j = 8;j < Length;j++)
        //数据赋值
    uchTemp[j] = uchData[i][j-7] * MAX_NUM/MAX_DEGREE;
    uchTemp[25] = CheckSum(uchTemp,26);
    .....
}
```

每台机器人都有属于自己的 ID 号，每个 ID 号就如同它们自己的名字。在上位机界面中，当我们输入 ID 号时，相应的机器人就会作出动作，这就实现了“一对一”的控制。那么对于一对多的控制，我们规定了机器人公用的 ID 号，当我们输入公用 ID 号时，所有接通电源的机器人都会受到控制，这就是所谓“一对多”的控制。本书中，使用五台双足步行机器人完成协调动作表演，站位如图 5-7 所示。

在完成了同时对双足步行机器人的控制后，而“千手观音”的动作表演往往是分时间差的，那么需要引进“延时”的概念，如图 5-8 所示。让前排的机器人先动起来，后排的逐个再动，整体效果就出现了“千手观音”表演的效果，如图 5-9 所示。

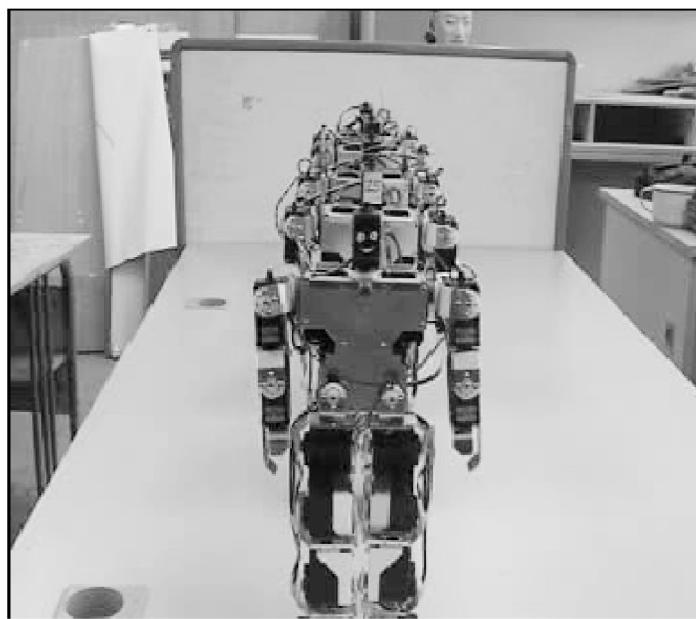


图 5-7 千手观音表演时的站位

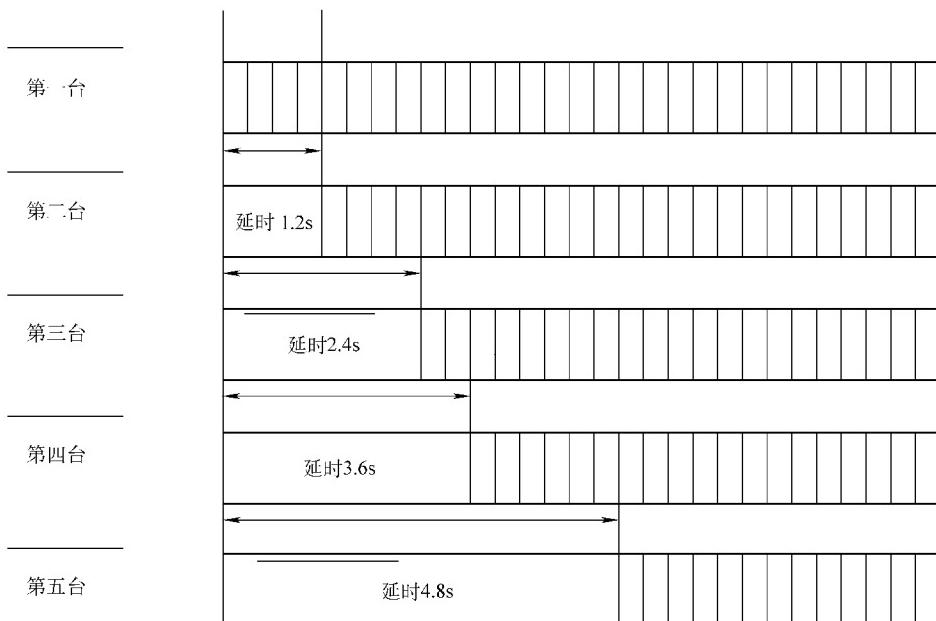


图 5-8 延时思想设计千手观音动作

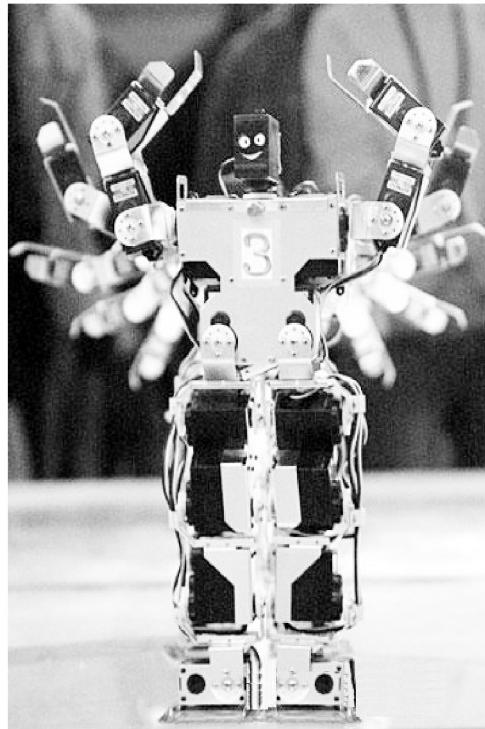


图 5-9 千手观音表演时的效果

5.2 双足步行机器人的情感模型、行为决策及情绪熵

5.2.1 隐马尔可夫模型

相对于情绪状态自发转移过程，情绪状态刺激转移过程则显得更加重要。根据心理学理论，在有外界情感信息刺激的情况下，情感状态主要体现在情绪的变化，而情绪的变化又受到多方面的影响，概括地说，主要由外界情感信息的刺激（类型、强度）、当前心情状态以及情感性格三个因素共同作用而发生影响，其基本的模型框架如图 5-10 所示。

隐马尔可夫链（Hidden Markov Model, HMM）是在马尔可夫链的基础上发展起来的。

由于实际问题比 Markov 链模型所描述的更为复杂，观察到的事件并不是与状态一一对应，而是通过一组概率分布相联系，这样的模型就称为 HMM。它是一个双重随机过程，其中之一是 Markov 链，这是基本随机过程，它描述状态的转移。另一个随机过程描述状态和观察值，不像 Markov 链模型中的观察值和状态一一对应，因此不能直接看到状态，而是通过一个随机过程去感知状态的存在及特性。因而称之为“隐” Markov 模型，即 HMM。

一个 HMM 可以由下面的五个参数描述：

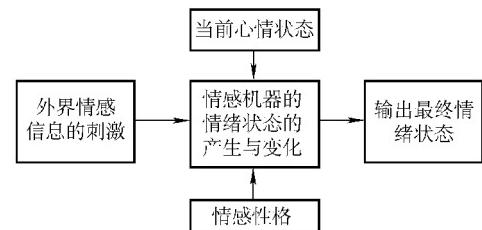


图 5-10 情绪状态刺激转移过程基本模型框图

N : 模型中 Markov 链状态的数目。记 N 个状态为 $\theta_1, \dots, \theta_N$, 记 t 时刻 Markov 链所处状态为 q_t , 显然 q_t 属于 $(\theta_1, \theta_2, \dots, \theta_N)$ 。

M : 每个状态对应可能的观察值的数目, 记 M 个观察值为 V_1, \dots, V_M , 记 t 时刻观察到的观察值为 O_t , 其中 O_t 属于 (V_1, \dots, V_M) 。

π : 初始位置概率矢量

$$\boldsymbol{\pi} = (\pi_1, \dots, \pi_N) \quad (5-6)$$

式中 $\pi_i = P(q_t = \theta_i) \quad 1 \leq i \leq N$

A : 状态转移概率矩阵

$$\mathbf{A} = (a_{ij})_{N \times N} \quad (5-7)$$

式中 $a_{ij} = P(q_{t+1} = \theta_j | q_t = \theta_i) \quad 1 \leq i, j \leq N$

B : 观察值概率矩阵

$$\mathbf{B} = (b_{jk})_{N \times M} \quad (5-8)$$

式中 $b_{jk} = P(O_i = V_k | q_t = \theta_j) \quad 1 \leq j \leq N, 1 \leq k \leq M$

由此, 可以记一个 HMM 为

$$\lambda = (N, M, \boldsymbol{\pi}, \mathbf{A}, \mathbf{B}) \quad (5-9)$$

或简写为

$$\lambda = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}) \quad (5-10)$$

更形象地说, HMM 可以分为两个部分, 一个是 Markov 链, 由 $\boldsymbol{\pi}$ 、 \mathbf{A} 描述, 产生输出状态序列; 另一个是随机过程, 由 \mathbf{B} 描述, 产生的输出为观察值序列, 如图 5-11 所示, T 为观察值时间长度。

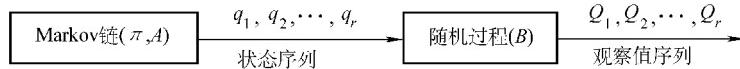


图 5-11 HMM 结构示意图

5.2.2 情感模型及行为决策的设计

对于双足步行机器人的感情来讲, 由于只有通过其动作表演来表达其个性, 所以相比带有表情机器人的机器人在情感的选择方面要狭窄得多, 所选择的情感通过实验论证, 只有“平静”、“喜悦”、“恐惧”、“愤怒”四个情感状态适合, 而其他的情感状态有待于进一步的研究探讨。

在双足步行机器人情感建模当中, 首先要寻找起始状态, 因为所有的过程都需要一个起点。通过对自然人的情感状态理解, 在没有外界刺激信号的作用下, 人一般都会平静, 对于双足步行机器人的情感设置也是一样的道理, 因此双足步行机器人的初始状态被定为“平静”状态。而其他状态都是与“平静”状态双向相通, 如图 5-12 所示。

那么其他状态之间的转化就需要通过平静状态来进行过渡, 而不允许随意两个状态之间进行转化。图 5-12 所示外周的状态都属于间接相连, 其中单弧形线表示这两个状态可以间接转化, 而双向箭头表示这两个状态可以直接转化。

基本的状态转化划分见表 5-1。

表 5-1 基本情感的状态转化

| 序号 | 状态转化 |
|----|----------|
| 1 | 平静—喜悦 |
| 2 | 平静—愤怒 |
| 3 | 平静—恐惧 |
| 4 | 喜悦—平静—愤怒 |
| 5 | 喜悦—平静—恐惧 |
| 6 | 愤怒—平静—恐惧 |
| 7 | 喜悦—平静 |
| 8 | 愤怒—平静 |
| 9 | 恐惧—平静 |
| 10 | 愤怒—平静—喜悦 |
| 11 | 恐惧—平静—喜悦 |
| 12 | 恐惧—平静—愤怒 |

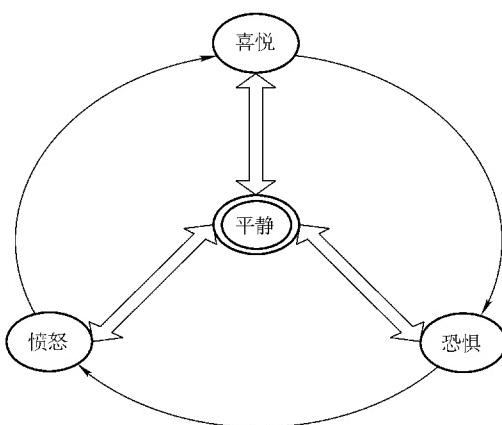


图 5-12 情感状态模型

而具体的情况，则要比上述的基本情感状态转化要复杂并具有随机性。由此通过将上述 12 种基本情感状态为元素，来进行任意的组合搭配，就能大致模拟现实中的情感状态转移的情况。

行为决策设计如图 5-13 所示。

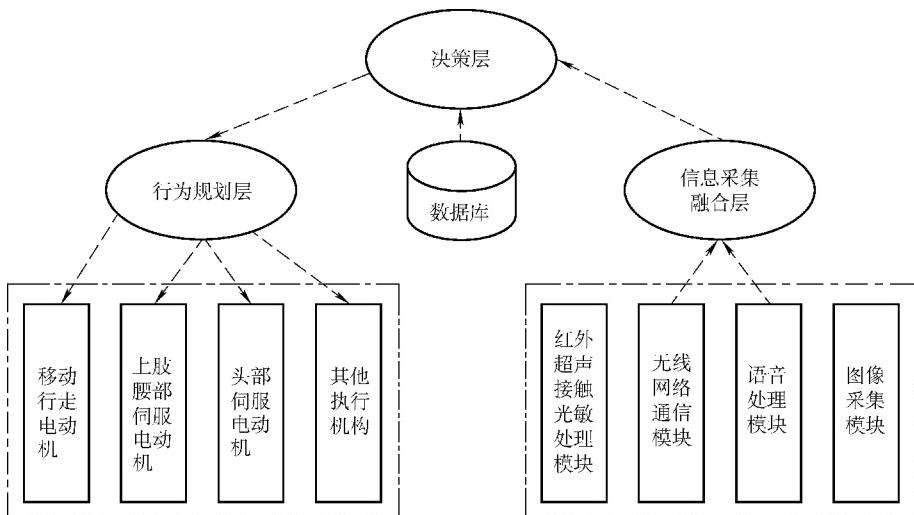


图 5-13 行为决策设计

一般来说，对于双足机器人的情感转移控制以及行为实现可以大致分为三部分：信息采集融合层、行为规则层、决策层。首先，信息采集融合层包含了几种外界刺激信号进入的途径，其中包括语音方面的语音处理模块，图像方面的图像采集模块，通信方面的无线网络通信模块和红外超声等模块。在这里主要介绍的信息采集是通过语音处理模块来实现的。决策

层的判断依据来自于数据库，然后通过数据库来连接行为规则层，最终连接到双足步行机器人的各个部分的伺服电动机。

由于语音数据库与情感方面的耦合还是处于初级阶段，数据库还无法随意地通过一句话而确定情感状态。例如：对于“今天天气不好”、“今天天气不错”等，语音数据库还无法直接通过里面的关键词来确定语句的褒贬，由此本章所介绍的语音数据库还处于人为设计阶段，也就是对应某个具体的语句，有相应的固定输出。

5.2.3 机器人情感转移控制的具体实现

情感决策模型的结构如图 5-14 所示，外界刺激事件通过感知评价转化为情感诱发变量，情感诱发变量分别作用于心情空间和情感空间，同时两者在个性因素的影响下，使得情感和心情状态发生转移，最后以合成表情做为情感状态输出。

在程序中，通过算法来做为实现情感状态的转移手段。情感状态可基本分为：平静、喜悦、愤怒、恐惧。那么很容易想到的是，将这四个状态分别对应一个变量 m_response，则可以得到下面的关系：

```
m_response = 0 //平静;
m_response = 1 //喜悦;
m_response = 2 //愤怒;
m_response = 3 //恐惧;
```

可以利用 switch 语句来设计条件限制。

执行动作程序的编写如下：

```
void CRobotSoftDlg::DoAction( unsigned int repeat, BYTE idnum, BYTE ActionNum ) //////
执行
```

```
{
    // TODO: Add your control notification handler code here
    unsigned char uchTemp[ 26 ];
    uchTemp[ 0 ] = 0xff;
    uchTemp[ 1 ] = 0xfe;
    uchTemp[ 2 ] = idnum;
    uchTemp[ 3 ] = 0x06; // 正常模式运行
    uchTemp[ 4 ] = ActionNum; // 

    for( int i = 5 ; i < 23 ; i + + )
    {
```

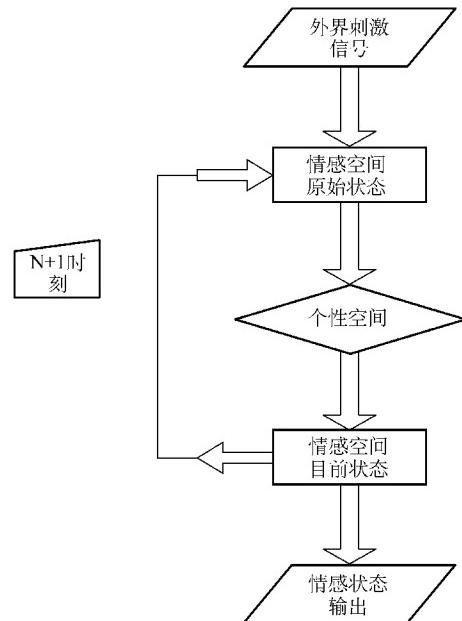


图 5-14 情感决策模型流程图

```

uchTemp[ i ] = 0x00;
}
uchTemp[ 23 ] = repeat%256;      //设定动作运行次数
uchTemp[ 24 ] = repeat/256;
uchTemp[ 25 ] = CheckSum( uchTemp,26 );
SendData( uchTemp,26,0 );
}

```

5.2.4 情绪熵的选取

情感是表征生命体心智状态的一种信息，将情感信息视为情绪过程产生的观察序列并假定这种情绪过程是一种马尔可夫过程，HMM 信号模型为描述情感信号处理过程提供了理论基础，从而可以获得期望的输出；其次，对情感信号建模可以更好地研究信号源——情绪过程，并且可以模拟信号源产生信号^[12]。如果拥有了这种情感信号模型，就能够预测信号将来最有可能出现的主特征值。由 HMM 模型中，可以得到心理状态的主特征趋势，并由此得到主特征值，那么如何区分呢？每个人的性格特点决定了他们的情感转移特点。不同性格的人，在这种情况下，从当前的情感状态向其他情感状态的转移速度快慢是不同的。因此，有必要引进情绪熵的概念，在程序中通过情绪熵的设置来控制机器人转移情感的速度快慢。

先了解一下熵的概念，在客观世界中，任一事物都是能发出信息的信源。而信源产生的信息带有随机性，即不定性。为了度量“不定性”，使信息定量化，1948 香农（Shannon）在《通信的数学理论》这一信息论的奠基性论文中，建立了定量“不定性”的函数式

$$H = -C \sum_{i=1}^n P_i \lg p_i \quad (5-11)$$

式中 H ——信源的平均信息量，也叫信息熵；

P_i ——信源第 i 种信号出现的概率；

C ——与对数的底及单位选择有关的常数。

香农的伟大贡献在于，利用概率分布的理论给出“熵”的严格定义。根据香农的定义，确定发生的事件如“太阳从东边升起”与确定不发生的事件如“太阳从西边升起”，其熵都是零。只有当发生与不发生的概率相同时，事件的熵才达到极大，其中所蕴涵的信息量达到最大。

基于以上讨论，由于情感是信息的一种，于是可以利用信息熵的方法去度量情感。我们来研究某生命体的 m 种情感，其中每种情感划分了 n 个级别，于是就具有了 n^m 个情感状态。令 $l = n^m$ ，那么情感空间中具有了 l 个情感状态，设 p_i ($i = 1, 2, \dots, l$) 为第 i 个情感状态的出现概率。

那么情绪熵就可以定义为

$$e_i = -C \sum_{j=1}^l P_{ij} \lg p_i \quad (5-12)$$

式中 e_i ——第 i 种状态的情绪熵；

P_{ij} ——第 i 种情感状态到第 j 种情感状态的概率；

C ——与对数底及单位选择有关的常数。

情绪熵是人类情感稳定性的度量尺度。就是说，当情绪熵的值越大，意味着情绪变化的倾向越不明确，表现为一种情绪的不稳定状态，而情绪熵的值越小，意味着情绪变化倾向越清晰明确，是情绪处于稳定状态的表示。所以，对于不同的人物，他们的各种心理状态下的情绪熵的最大值反映了他们的性格特征。就是说，利用情绪熵，可以构造出不同个性的机器人的感情。

在这里引用情绪熵的概念并进行相应简化，以方便模拟机器人的内外向的性格特征，这里规定情绪熵越小，机器人的个性趋向于内向；情绪熵越大，机器人的个性趋向于外向，见表 5-2。

表 5-2 情绪熵与动作循环次数的转化表

| 情绪熵 | 动作循环次数 | 情绪熵 | 动作循环次数 |
|-----|--------|-----|--------|
| 1 | 3 | 4 | 12 |
| 2 | 6 | 5 | 15 |
| 3 | 9 | 6 | 18 |

将上述的表达通过程序流程图来表示，如图 5-15 所示。

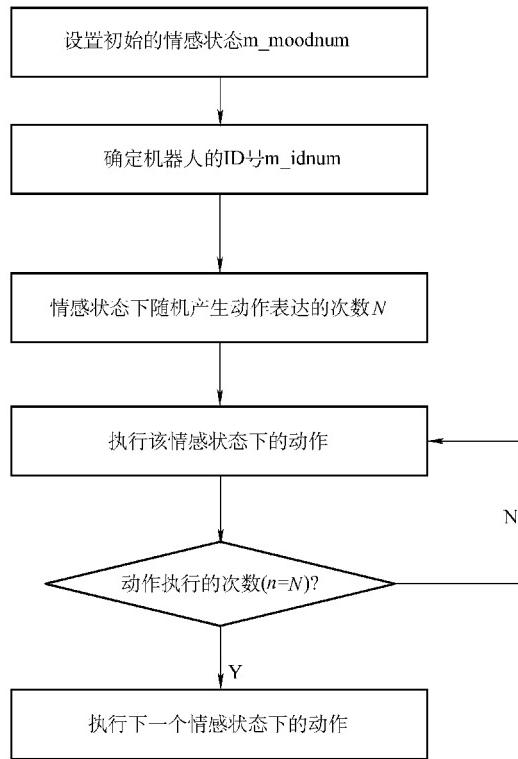


图 5-15 具有情绪熵的情感状态转移的程序流程图

由此可知，情绪熵的选取具有很重要的意义，它决定了个性空间设计的效果。在双足步行机器人的情感转化与行为决策的实现方面，情绪熵的选择给机器人添加了性格内外向的特征，使得双足步行机器人在人工心理与人工情感领域的应用中，更接近人的性格品性。

5.3 双足步行机器人的语音控制

5.3.1 语音在双足步行机器人控制平台上的实现

双足步行机器人上位机语音识别程序框架：

在程序的开始，首先要利用语音(Software Development Kit)SDK提供的Asr_SetSystemDir()函数设置识别引擎工作的系统目录，主要是指定声学模型所在的目录，然后利用Asr_SysInit()初始化语音识别系统。

如果语音识别系统初始化成功，则可以利用Asr_LineInit(ResultProc)函数来创建一个识别引擎hLine，并指定该引擎识别结果出来时的回调函数以及用户定义数据。

在本文中我们回调函数的定义为

```
DWORD ResultProc( HASRLINE hLine, const PASR_NOTIFY pNotify, DWORD dwUserData);
```

其中pNotify是回调通知，它的结构体如下：

```
typedef struct _ ASR_NOTIFY
{
    int NotifyType;
    int data;
    DWORD dwReserve;
} ASR_NOTIFY, *PASR_NOTIFY;
```

这个结构在回调函数中返回和语音以及识别结果相关的数据，其中NotifyType为通知类型，包括：

| | | |
|--------------------|---|---------|
| NOTIFY_RESULT | 1 | //识别结果 |
| NOTIFY_VOICE_LEVEL | 2 | //声音大小 |
| NOTIFY_UTT_BEGIN | 3 | //一句话开始 |
| NOTIFY_UTT_END | 4 | //一句话结束 |

data：当NotifyType为NOTIFY_VOICE_LEVEL时，data表示当前录音的大小，NotifyType为其他值时无意义。

dwReserve：保留，无意义。

成功创建一个识别引擎后，利用Asr_GrammarLoad()函数来导入语法规则文件中的语法规则，并对语法规则进行设置。语法规则文件可以由用户按照语法规则的要求依据需要自己编写，语音识别所需要用到的语法规则，遵循ABNF范式(Augmented BNF)，兼容W3C语法定义。语法规则的结构体如下所示：

```
typedef struct _ GRAMMAR_ACTIVATE
{
    int nNum;
    char * szRuleName[ 100 ];
} HASR_GRAMMAR, *PHASR_GRAMMAR;
```

我们可以使用该结构来设置、切换、关闭识别引擎对应的语法规则。其中：
nNum 是语法规则的数量。

szRuleName 是语法规则的名字，一个语音识别引擎可以设置多个语法规则。

成功导入语法规则后，我们利用 Asr_GrammarActivate() 函数来激活语法规则，设置 Asr_RecSetAutoWork(true) 由引擎内部自动采集语音数据，调用 Asr_SetParam() 设置录音以及引擎相关的参数，调用函数 Asr_RecStartX(hLine) 开始自动识别。

端点检测到语音大于系统设置的起始话音音量并进行录音，当语音小于设置的结束音量停止录音。语音识别系统对语音进行识别，通过回调函数得到回调通知，利用 Asr_GetNBestResult() 获得识别结果。

如果不再需要语音识别，利用 Asr_RecStopX(hLine) 函数停止由 Asr_RecStartX 函数开启的识别任务，利用 Asr_LineRelease(hLine) 关闭 hLine 语音识别引擎，并且释放其相关的资源，利用 Asr_SysRelease() 退出系统，清除识别系统申请的资源。

安装语音合成软件 InterPhonic CN Rev1.0 版，安装 iFlyTTS_SDK，安装加密狗配置程序（在安装盘中）。

数据源：在系统 DSN 里注册数据源。

驱动：Microsoft access driver(*.mdb)。

数据库：WuKongdb.mdb。

在系统 DSN 中注册数据源，打开控制面板→管理工具→数据源(ODBC)→系统 DSN→添加→选择 Microsoft access driver(*.mdb) 点击“完成”后→在数据源名里输入 dashendb 同时在数据库(选择)中选择相应路径下 WuKongdb.mdb 即可。

把串口 1 和无线发射器连接起来。

把 Mandarin_8KHz3.0、asrapi.dll、ASRAPI.h、asrapi.lib、Dashen.grm 文件复制到当前目录下，在 vc++ link 库中加入 asrapi.lib，在工程中加入 ASRAPI.h 文件。

编写语法规则：新建一个记事本命名为 Dashen.grm，在里面写入：

```
$ talk = (* * * | ...) <0 -> $ talks(* * * | ...) <0 ->;
```

```
$ talks =
```

```
你好！
```

```
你会唱歌吗！
```

```
你多大了；
```

Dashen.grm 文件增加语法的规则：“\$ talks =”后面增加要识别的语句需顶行开始，以“|”结尾，文件的最后一条语句以“;”结尾。否则不能进行识别！

```
#include "stdio.h"  
#include "iostream.h"  
#include "asrapi.h"  
#include "windows.h"  
#define ONLINE_TEST
```

语音控制平台中的重要函数编写如下：

```
//-----在数据集中查找所提问题的答案-----  
Dashen CRobotSoftDlg::FindFunctionData(CString str)
```

```

{
    CString str1,str2;
    Dashen Ds1;
    str1 = str;
    str1. TrimLeft( );
    str1. TrimRight( );

    //MessageBox( str );

    int Recordcount = 0;
    int number;
    CRecSetFunct FucSet;

    FucSet. Open( );

    while( ! FucSet. IsEOF( ) )
    {
        str2 = FucSet. m _ Question;
        str2. TrimLeft( );
        str2. TrimRight( );
        FucSet. m _ Question. TrimLeft( );
        FucSet. m _ Question. TrimRight( );
        // for( int j = 0; j < str1. GetLength( ); j + + )
        // {
            //if( str1. Mid( j,str2. GetLength( ) ) == str2 )
            if( str1  == str2 )
            {
                //MessageBox( str2 );
                number = FucSet. m _ ID; //储存符合条件的每条记录的 ID 号
                Recordcount + +; //查找到的符合条件的记录数,每查找到一条值自动加一
                FucSet. MoveLast( );
            }
        // }
        FucSet. MoveNext( );
    }
    if( Recordcount == 0 )
    {
        Ds1. Answer = "我没听清楚你说的,请再说一遍!";
    }
}

```

```

Ds1. Act = "0000" ;

//MessageBox( "我没听清楚你说的" );
}

else

{
    srand( ( unsigned )time( NULL ) ); // 

    CString ch;
    ch.Format( "%d" , number ); //把整数转换为 Cstring 类型

    FucSet. m _ strFilter = "ID = " + ch; //查找表中 ID 等于 ch 的记录,实现在所有符合
    条件的记录中随机调出一条的功能

    FucSet. Requery( );
    Ds1. Answer = FucSet. m _ Answer;
    Ds1. Act = FucSet. m _ Action;

    //MessageBox( Ds1. Act );

    FucSet. Close( );
}

return Ds1;
}

//-----在数据集中查找所提问题的答案-----

```

5.3.2 语音数据库的设计

在语音数据库中，首先要编写许多具有跟情感相关联的语句，通过这些语句的输入，即作为外界刺激信号，使得程序在数据库中直接匹配该语句的情感状态。

表 5-3 所示为语音数据库简单示例。

表 5-3 语音数据库设计表

| 语句 | 对应的情感状态 | 对应的字符串 |
|--------|---------|--------|
| 今天天气真好 | 喜悦 | 01 |
| 你好像没电了 | 愤怒 | 02 |
| 你真聪明 | 喜悦 | 01 |
| 你怕关电源吗 | 恐惧 | 03 |
| 天是蓝色的 | 平静 | 00 |
| 你真漂亮 | 喜悦 | 01 |

(续)

| 语句 | 对应的情感状态 | 对应的字符串 |
|---------|---------|--------|
| 你怕摔吗 | 恐惧 | 03 |
| 你是大明星 | 喜悦 | 01 |
| 我不认识你 | 愤怒 | 02 |
| 老虎来了 | 恐惧 | 03 |
| 你跳舞真好 | 喜悦 | 01 |
| 领导来看演出了 | 喜悦 | 01 |
| 你太鲁莽了 | 愤怒 | 02 |
| 地球是圆的 | 平静 | 00 |
| 你该退休了 | 恐惧 | 03 |

这里面的对应字符串包含着情感信息，在程序中用 m_response 来存储，而这里面对应的多智能体控制来说，还需要有 m_idnum 来存储机器人的 ID 号，以确保这个命令是发给具体的某个机器人的。

在实验中，运用机器人的个数决定了存储机器人 ID 号的变量个数，以两个机器人为例，分别对应 ID 号“01”“02”，那么在对应的字符串中就应该添加上 ID 号的信息。

语音数据库的字符串的结构见表 5-4。

表 5-4 语音数据库字符串结构表

| 第一个机器人 ID 号 | 第二个机器人 ID 号 | 外界刺激信号反映的情感 |
|-------------|-------------|-------------|
| 01 | 02 | 01 |

上述表示一号机器人与二号机器人在外界刺激信号作用下，所确定的喜悦的情感状态。

如何处理这个数据库转过来的字符串尤为重要，它是这就预示着整个实验的成败。将字符串“010201”进行处理，然后再将变量赋值。程序编写过程如下：

```
////////////////////////////wangtianzeng//////s 读取文件发送动作数据并运行/////////
```

```
void CRobotSoftDlg::ReadFromFileAndRun( CString str )
{
    int iStrlength;
    iStrlength = str.GetLength();
    if( iStrlength != 6 ) //数据库返回的字符串长度为 6
    {
        MessageBox( "文件格式不正确,请检查数据库中的动作文件格式!" );
        return;
    }
}
```

```
CString str1,str2,str3
```

```
str1 = str.Left(2); //取第一个机器人的 ID 号
```

```

str2 = str. Mid(2,2); //取第二个机器人的 ID 号
str3 = str. Right(2); //取情感状态

m_idnum1 = (unsigned char)_tcstoul(str1, 0, 16); //将字符串型转化成整形
m_idnum2 = (unsigned char)_tcstoul(str2, 0, 16);
m_response = (unsigned char)_tcstoul(str3, 0, 16);
DoAction1(1, m_idnum1, m_response); //执行第一个机器人的动作命令
DoAction2(1, m_idnum2, m_response); //执行第二个机器人的动作命令
}

///////////////////////////////wangtianzeng//// e 读取文件发送动作数据并运行///////////

```

5.4 双足步行机器人动作的设计

5.4.1 基于情感方面的动作设计

双足机器人的情感由上一节描述应该分为大致平静、喜悦、愤怒、恐惧四种状态，那么在这四个状态中应该分别对应有能够表达其情感状态的机器人的动作。

由于我们试验平台的双足机器人不同于具有表情的大型拟人机器人，因此表达其情感的途径就只有它的动作。动作的设计要符合人的正常行为规范，如果动作设计得太夸张就失去了“类人”的意义，换句话说，让双足机器人做出人无法做到的动作是没有意义的；如果动作设计得太“温和”，即幅度小，这样虽然具有模仿人动作的作用，但无法清晰地表达双足机器人的情感，这样的动作在本书中不予采纳，但可以在其他方面运用，例如招手、伸展运动等。

那么情感——动作关系表就应该见表 5-5。

表 5-5 情感——动作关系表

| 情感状态 | 双足机器人的动作设计 |
|------|------------|
| 平静 | 直立 |
| | 踮脚跳 |
| | 挥臂跳舞 |
| | 单腿支撑跳舞 |
| | 倒立跳街舞 |
| | 鞠躬 |
| 喜悦 | 打拳 |
| | 踢腿 |
| | 做俯卧撑 |
| | 侧走甩臂 |
| 愤怒 | 扩胸运动 |
| | |

(续)

| 情感状态 | 双足机器人的动作设计 |
|------|------------|
| 恐惧 | 下蹲 |
| | 手遮脸 |
| | 蹲下抱头 |
| | 摇摆 |
| | |

平静状态下的动作如图 5-16 所示，喜悦状态下的动作如图 5-17 所示。

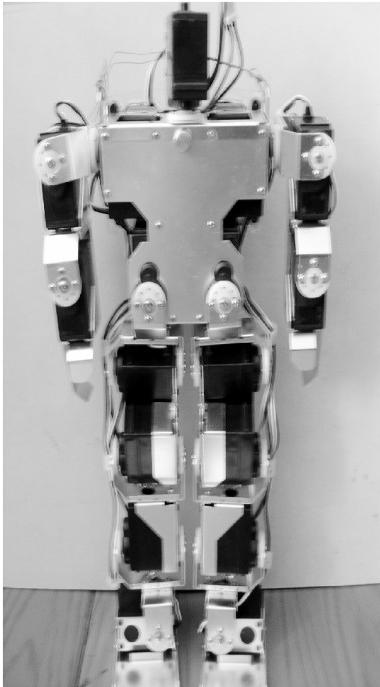


图 5-16 平静状态下的动作



图 5-17 喜悦状态下的动作

5.4.2 基于性格方面的动作设计

对于个性空间，通过设置不同的情绪熵，来设置机器人对于情感状态转移的难易度，这从外表看来就如同人一样，对于情感状态转移比较容易的机器人，它很容易受到外界刺激的影响，表现很活泼，如同性格外向的人一样，反之就是内向性格。

通过动作的频率快慢也可以体现机器人的性格特点，动作频率快往往能表现机器人的外向特点，这在起初的对于具体的机器人的动作设计时就可以设置，这里有具体负责动作速度的设置，如图 5-18 所示。

其中的速度控制影响伺服电动机转动的速度，速度 7 为最大，适合性格外向机器人的动作设计，而速度 1 为最小，适合性格内向机器人的动作设计。由此通过舵机转动组成动作元素，再组成基本动作，然后组成复杂动作，最后表现为机器人的行为，如图 5-19 所示。



图 5-18 上位机中动作速度的设置图

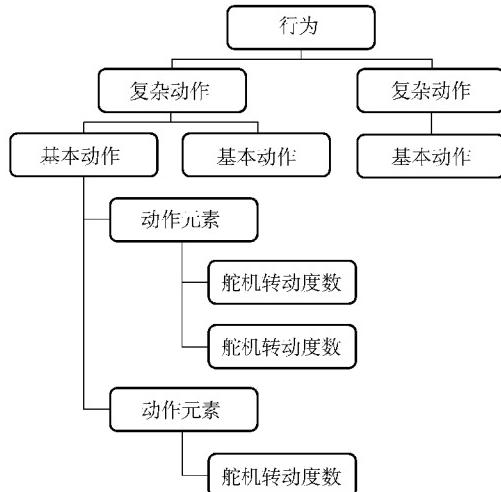


图 5-19 机器人行为结构图

5.5 小结

本章结合第 2 章所介绍的机器人控制的相关理论与技术，着重地对双足机器人的情感建模与行为实现进行设计和完成。在 VC + + 开发环境和已有的无线通信模块的基础，对多智能体控制中的动作协调设计、动作转化程序以及对复杂的“千手观音”机器人动作表演进行分析和介绍。为了使双足机器人具有一定的情感与情绪，通过隐马尔可夫模型的建立，完成了对双足机器人情感建模以及行为决策的设计，实现了情感转移控制。同时介绍了情绪熵的概念，使得机器人具有一定的个性化，即内向或外向的性格取向。将传统的手动操作机器人界面发展成为语音对双足机器人的控制平台，进而设计出相应的语音数据库，将具体对话内容与情感状态进行有机关联。对机器人动作设计进行优化，使得动作在表达机器人的情感方面、性格方面都表达得更加合理化，以达到在机器人实物课题实验演示中，取得更好的效果。

参 考 文 献

- [1] 李娜. 用于情感交互的机电综合平台研究 [D]. 北京: 北京科技大学, 2006.
- [2] <http://world.honda.com/ASIMO/>
- [3] 国家 863 计划智能机器人专家组. 机器人博览 [M]. 北京: 中国科学技术出版社, 2001.
- [4] Breazeal C. Function meets style: insights from emotion theory applied to HRI Systems, Man and Cybernet-

- ics, Part C, IEEE Transactions 2004, 34 (5): 187-194.
- [5] Zadeh A. From Computing with Numbers to Computing with Words: From Lotfi Mumpulation of Measurements to Manipulation of Perceptions [C] //IPMM'99, Honolulu, Hawaii, 1999.
- [6] 蔡自兴, 徐光信. 人工智能及其应用 [M]. 北京: 清华大学出版社, 2003.
- [7] 王志良. 人工心理学——关于更接近人脑工作模式的科学 [J]. 北京科技大学学报, 2000 (5).
- [8] 王天增. 基于类人机器人情感建模和行为决策的研究 [D]. 北京: 北京科技大学, 2007.
- [9] 范辉, 李晋江, 张晖. Agent 技术的研究现状与发展趋势 [J]. 微型机与应用, 2001 (10).
- [10] 国家自然科学基金委员会. 神经科学 [M]. 北京: 科学出版社, 1997.
- [11] 张帅, 卢昱. 一种基于多智能体分布式控制系统结构的受控网络 [J]. 装备指挥技术学院学报, 2004, 15 (4): 77 ~ 81.
- [12] 斯托曼 K T. 情绪心理学 [M]. 张燕云, 译. 沈阳: 辽宁人民出版社, 1987.

第6章 双足机器人情感化步态

6.1 引言

随着当今人工智能的发展，我们对双足机器人的要求已不仅仅是完成机械作业和逻辑推理，简单地模仿人的步态行动，以及如何让机器人更加智能化、赋有情感，是目前的研究趋势之一。双足机器人更高阶的应用之一就是类人型机器人，更易于适应人类环境并为人类所应用，例如在服务业和娱乐产业中，它们可以辨识更精准的语言、表情、手势等信息并加以分析，解读出在复杂环境下人类多变的情感变化，形成更具备人类形态的姿势动作。本章将着重研究如何赋予双足机器人情感使其能够在不同情感状态下走出不同的步态。

肢体动作是情感的外化，将各类情感寓于各种不同的步态行为中，可以表达出更精确的情感信息，这种表现方式直观且形象。我们在建立准确的情感模型之前，先要归纳在不同的情感状态下相对应的步态表现。

人类的情感状态除了平静以外，一般有喜悦、悲伤、拒绝、厌恶、激动、愤怒等。因此双足机器人在不同情感状态下可以拟设计为兴奋时双足机器人会双腿跃起，显示出一种很开心的状态，行走时速度会加快，跨步较小；悲伤沮丧时，它会步伐缓慢迟滞，跨步明显变小；当表达拒绝或厌恶的情感时，双足向后退，并站立，身体明显向后倾等。

现今对情感化的双足机器人的研究越来越引起研究者们的广泛关注，在有关机器人的研究里逐渐变为热门研究对象，其中将机器人赋予肢体语言并使其表达自身的情感，成为机器人研究领域的一个新方向。

6.2 情感模型相关理论

现如今，对情感进行度量的思想吸引着心理学、认知科学和信息科学等很多学科的研究人员。由于情绪的复杂性和对人类本身情感变化规律认知的不完善性，因此虽然已有不同学科的研究者从不同角度试图模拟情绪的产生和变化，但这项工作依然十分艰巨。而情感建模在功能的角度实现了有限的模仿，是机器情感化的一大重要突破。国内外研究者针对情感的相关有效算法，提出了在一定范围内的不同模型用来进行类似情感的计量，如知名的 OCC 算法、Kismet 情感模型、HMM 情感模型、基于欧式空间的人工情感数学模型等，具体如图 6-1 所示。

OCC 情感模型是 1988 年 Ortony、Clore 和 Collins 在所出版的 *Cognitive Structure of Emotions* 一书中给出的著名情绪认知模型。它是第一个以计算机实现为目的而发展起来的模型。它通过分析环境中所发生的各种事件，以及和其他实体交互的行为可能发生的各种情感，在此基础上以规则的形式总结并归纳出这些对应关系。OCC 模型根据不同的情感刺激提出 22 种情感，同时对产生每种情感的刺激特征进行了详细描述。它是一种基于情绪的认知理论和

规则的建模，对于情绪的激活和产生更多考虑的是认识的因素，没有涉及情绪激活的其他过程。

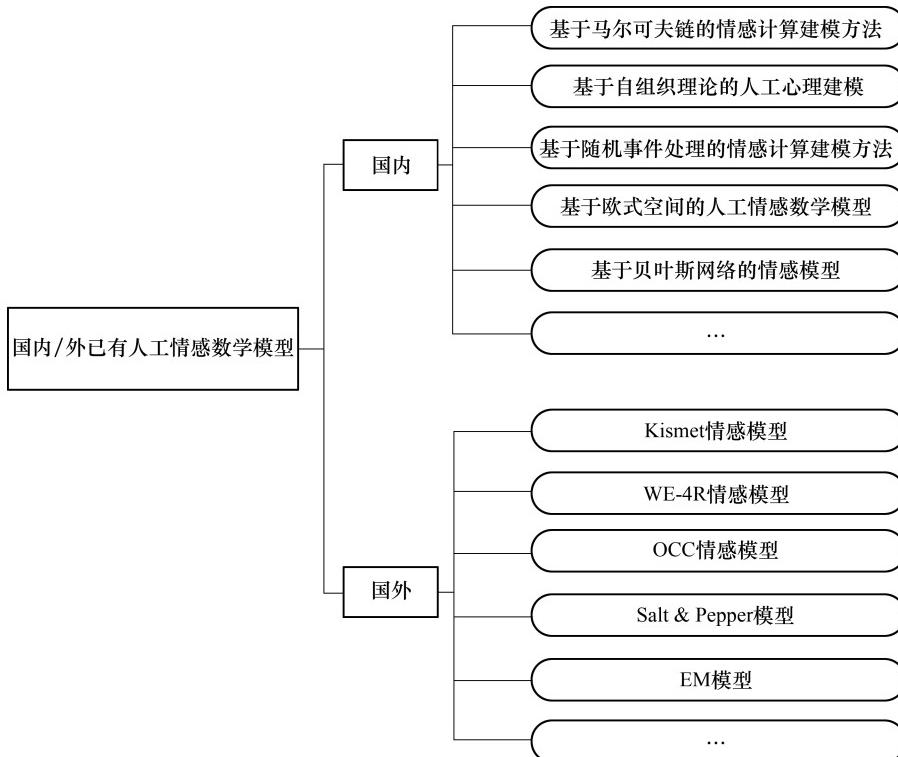


图 6-1 国内外已有人工情感数学模型

Kismet 情感模型是作为环境、内部刺激和行为动作的媒介，通过这个情感模型，机器人对外界输入的刺激和内部需要进行综合判断，从而引起表现行为的各种变化，机器人能够表达愤怒、高兴、兴趣、惊恐及平静等情感。此模型可以分为情绪刺激、情绪评价、情绪激活、情绪表达四个部分。外界情感作用于该情感模型时，该情感模型运用效价、唤醒度及姿态这样的表现形式识别作用于它的激励，之后传递到情感空间，来激活情感。Kismet 模型框架如图 6-2 所示。

美国研究人员 Ekman 和 Friesen 发现，人类有 6 种基本情绪，即快乐、悲伤、恐惧、惊讶、愤怒、厌恶。这 6 种情绪可以相互组合，并派生出其余各种各样的复合情绪，如忧郁、紧张、焦虑等。此 6 种情感在文化传中的差异性小，其他情感都能够由这 6 种基本情感构成。以这 6 种情感作为基本情感，每种个性化的情感表现都能够表示为一个具有 6 维的向量，在 t 时刻 ($t > 0$) 每种情感所具有的情感强度可以由以下公式得到：

$$E_t = (E_{\text{happy},t}, E_{\text{sadness},t}, E_{\text{anger},t}, E_{\text{surprise},t}, E_{\text{fear},t}, E_{\text{disgust},t})$$

其中， $E_{\text{happy},t}$, $E_{\text{sadness},t}$, $E_{\text{anger},t}$, $E_{\text{surprise},t}$, $E_{\text{fear},t}$, $E_{\text{disgust},t}$ 是情感状态在 t 时刻的从高兴到厌恶的情感强度值。

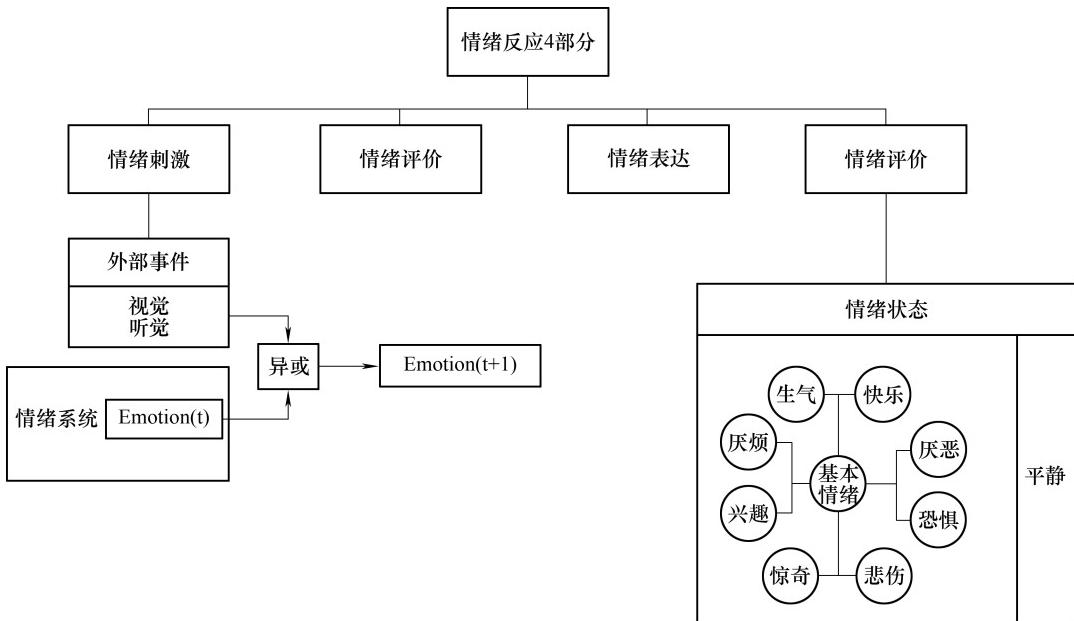


图 6-2 Kismet 模型框架图

6.3 基于欧式空间的人工情感模型及 Ekman 情感理论

1. 基于欧式空间的人工情感模型

如图 6-3 所示的欧式情感空间，是将基础情感作为基矢量，基于情绪多维量结构的维度空间，将情感能量化的概率空间。在此基础上研究机器情感的组成和情感状态变化的数学描述，利用基于欧式空间的情感建模方法，研究模型参数与个性心理之间的关系。

欧式情感空间模型内离散的点都是由基础情感组成的，如图 6-3 所示的空间包括了 3 种基础情感，即喜悦、恐惧和愤怒。而情感空间的其他点是上述 3 种基础情感按轻重程度不同以不同方式组合而成的。例如，规定情感状态的每一维仅取 0, 0.5, 1 这 3 个数值，而喜悦就可以分为喜悦 (0), 喜悦 (0.5), 喜悦 (1) 三个状态。于是这样的可列举情感状态共有 27 个状态点。其中，原点 (0, 0, 0) 除了代表平静外还可以代表其他未知的情感状态。当情感模型中只有这 3 种基础情感状态时，称为完全情感空间，此时原点为平静点；当情感模型中不只有这 3 种情感状态时，称为不完全情感空间，此时原点有可能是平静点，也有可能是其余没有包含在这个情感模型中的情感状态。

情感活动可以看成这个情感空间中各状态转移的过程，即马尔科夫过程。一般来讲，如果情感空间中包括不止 3 个基础状态，还包括了 n 个情感状态，每个情感状态可以分为 m 个等级，即 $l = m^n$ ，这便是 l 维马尔科夫概率矩阵。

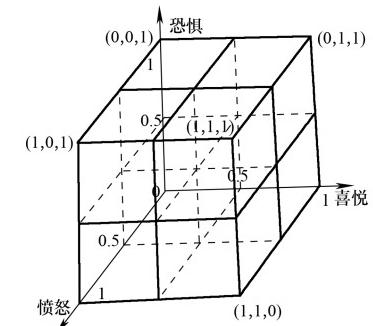


图 6-3 欧式情感空间

2. HMM 情感模型

HMM 情感建模给出了情感能量、情感强度和情感熵等概念来描述情感特征和情感状态。通过计算仿真，验证所提出模型可以较为正确地模拟情绪状态的自发转移及在外界刺激时转移、变化的动态过程，完整地描述情绪强度在外界刺激、当前心情状态和性格 3 个方面因素的综合影响下的变化规律，为情感计算和机器情绪自动生成理论研究提供一种崭新的方法。

由于情感是信息的一种，因此可以用信息熵的方法来度量情感。当情绪有 n 个状态，每个状态分为 m 个等级时，就有了 $l = m^n$ 种情感状态。设 P_i ($i = 1, 2, \dots, l$) 为第 i 个情感状态出现的概率，情感熵 E 就可以定义为

$$E = -C \sum_{i=1}^l P_i \log P_i \quad (C \text{ 为与对数底及单位选择有关常数})$$

如果情感空间中每一个情感状态出现的概率均相同时，那么 $P_i = \frac{1}{m^n}$, $i \in (1, 2, \dots, m^n)$ ，此时情感熵达到最大值。

$$E_{\max} = -C \log P_i = C n \log m$$

E_{\max} 代表研究范围内生命体情感复杂程度的最大值，也就是说， E_{\max} 是上界，越接近此数值意味着感情越丰富。

在情感空间的 m^n 个情感状态中，设某一个时刻处于第 i 个状态，那么它到其他状态的概率就构成了一个情绪的概率矢量

$$\vec{e}_i = (p_{i1}, p_{i2}, \dots, p_{ii}, \dots, p_{il}), l = m^n, \sum_{j=1}^l p_{ij} = 1, i \in (1, 2, \dots, l)$$

此时情绪熵就定义为

$$e_i = -C \sum_{j=1}^l p_{ij} \log p_{ij}$$

式中 e_i ——在第 i 种状态的情绪熵；

p_{ij} ——第 i 种情感状态到第 j 种情感状态的概率。

6.4 双足机器人步态情感模型的建立

本节将结合 HMM 情感模型与 Baum-Welch 算法，应用于双足机器人步态的情感模型建立。

1. 情感模型的传递

机器人整体传递系统由两大类组成，即输入端和输出端，具体如图 6-4 所示。输入端是通过情感模型空间对外界刺激进行反应，输出端是机器人通过情感能力系统进行步态姿势输出。整体构成了机器人情感接收和反应的过程，包括情感因子的提取和分析。除此之外还有外界情感刺激，情感模型与步态姿势的对应等。

2. 基于 HMM 的情感建模

我们首先要确定情感的每个层次上的种类数量，即 HMM 中对应的状态数 n 和心情种类对应的个数 m 。在每种心情中，各种状态以一定概率出现。HMM 由两部分组成，一个为马尔可夫链，由 π 和 A 描述（ π 为初始状态概率矢量， A 为状态概率转移矩阵），它们决定了马尔可夫链的形状。其中，马尔可夫链为状态遍历类型，严格来说，遍历模型的性质是任何

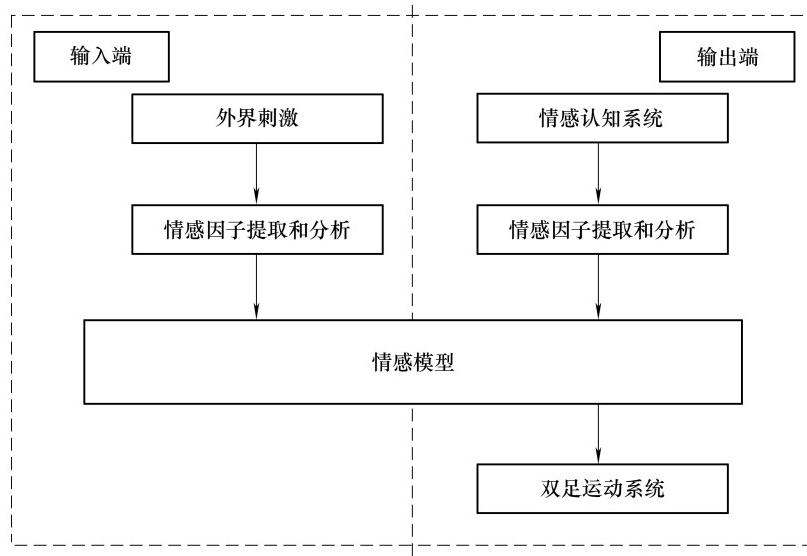


图 6-4 情感模型的传递

一个状态可由任一个其他状态在有限步内到达。此类模型性质为每个 a_{ij} 系数都是正的，即对任意 i, j ，有 $a_{ij} > 0$ 。HMM 另一组成部分由 \mathbf{B} 描述， \mathbf{B} 为观察概率矩阵，产生的输出为观察值序列。HMM 的组成如图 6-5 所示。

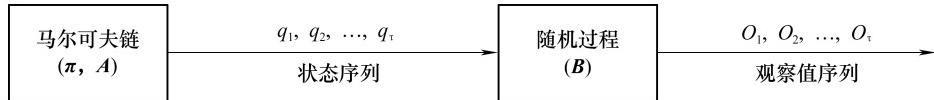


图 6-5 HMM 组成示意图

情感熵和情绪熵对 π 和 A 的选取也会产生影响，对于某个人而言，在外界刺激变化跨度不大时，其对应某一心理状态的情绪熵数值应基本相等，所以我们假定在 3 种情感状态（平静、开心、难过）时，给出矩阵 A_R 和 A_P (A_R 为开心时的状态概率转移矩阵， A_P 为难过时的状态概率转移矩阵)。建立矩阵时要考虑 3 点约束：

- 1) 符合人类情感规律；
- 2) 符合 HMM 中马尔可夫链的约束条件，对任意激励均有 $\sum_{j=1}^N a_{INkj} = 1$ ；
- 3) 要使由 HMM 构造的个体情绪熵符合个体性格特征。

我们假定当双足机器人未受任何激励作用时的初始概率矢量 π 为 $(1, 0, 0)$ 。我们把外界刺激信号分为导致开心结果的刺激和导致难过结果的刺激两种。这样针对每种刺激分别有一个状态转移概率矩阵 (A_R 和 A_P) 及可见符号概率矩阵 (B_R 和 B_P) 与之对应。用于模型重估的状态转移概率矩阵及可见符号概率矩阵初始值的选取，可以由经验来决定，同时考虑外界刺激对各个心态下情绪熵的作用。

在接收到开心的刺激时，状态转移概率矩阵为

$$A_R = \begin{bmatrix} 0.29 & 0.7 & 0.01 \\ 0.09 & 0.9 & 0.01 \\ 0.4 & 0.5 & 0.1 \end{bmatrix}$$

$$\mathbf{B}_R = \begin{bmatrix} 0.3 & 0.6 & 0.1 \\ 0.149 & 0.85 & 0.001 \\ 0.651 & 0.2 & 0.149 \end{bmatrix}$$

在接收到难过的刺激时，状态转移概率矩阵为

$$\mathbf{A}_P = \begin{bmatrix} 0.29 & 0.01 & 0.7 \\ 0.4 & 0.1 & 0.5 \\ 0.09 & 0.01 & 0.9 \end{bmatrix}$$

$$\mathbf{B}_P = \begin{bmatrix} 0.3 & 0.1 & 0.5 \\ 0.651 & 0.149 & 0.2 \\ 0.149 & 0.01 & 0.85 \end{bmatrix}$$

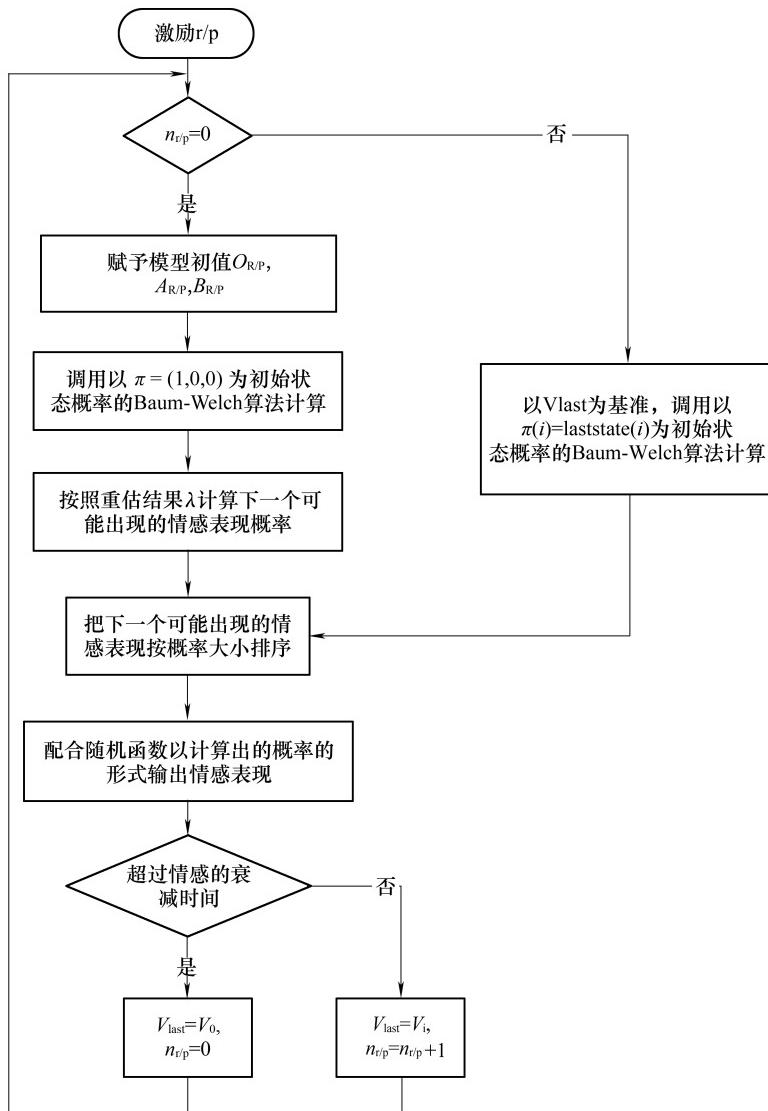


图 6-6 基于 Baum-Welch 算法的程序系统

为了得到充足数据对所有参数进行可靠估计，可以采用多重观察序列，即在实验中多次激励双足机器人，记录在不同激励下得到的双足机器人步态表现，用特定值来表示特定结果。观察序列用二维数组 $o(l, t)$ 表示， l 代表第 l 个观察序列， t 代表观察时刻。此步骤用来调整初始参数，得到给定模型的观察序列概率最大的初始模型参数 $\lambda = (A, B, \pi)$ 。

3. 建立结合 Baum-Welch 算法的程序系统

利用 Baum-Welch 算法计算情感模型参数的过程为，给出一组训练数据，经过算法程序的运算，可以得到使给定观察序列出现概率局部最大的模型参数估计结果，从而为情感模型参数的确定提供一种理论分析方法。根据刚才提供的初始值和实验中的观察序列训练模型，由最后的模型产生观察序列并且综合考虑行为模式因素和输入信号判断因素的影响，得到如图 6-6 所示流程图。

6.5 基于情感的双足运动系统

本节将以日本早稻田大学仿人机器人研究所研发的机器人 WABIAN-RII 为例，描述基于情感的双足机器人运动系统。在本节中，考虑 3 种基本情感，即平静、快乐和难过。这些情绪通过双足机器人行走风格表示，行走风格通过全身运动化参数预先设定。为了在情绪表达期间保持机器人平衡，采用躯干运动辅助平衡，其通过基于头部，手臂和腿部运动的补偿运动控制来计算。

双足类人形机器人具有 43 个机械自由度和 4 个被动自由度（平移运动和 3 个旋转运动）。WABIAN-RII 如图 6-7 所示，其高度约为 1.84m，总重量为 127kg。

1. 运动参数和行走模式

(1) 平静状态下的行走模式 在平静状态下，双足机器人采用平滑模式行走。腰、头部、脚可以由多项式决定。平滑行走需要考虑角度、角速度、角加速度和脚最高位置的角度，通过使用 6 阶多项式生成地面参数。通过选择初始值和最终值产生脚位置和方向的 3 个约束

$x_{\text{foot}}(t_0) = x_0, x_{\text{foot}}(t_f) = x_f, x_{\text{foot}}(t_m) = x_m$
式中， t_0 为起始时间， t_m 为中间时间， t_f 为终止时间。

位置和取向 x_{foot} 还有 4 个约束，即零初始速度和加速度、最终速度和加速度

$$\dot{x}_{\text{foot}}(t_0) = \mathbf{0}, \dot{x}_{\text{foot}}(t_f) = \mathbf{0}, \ddot{x}_{\text{foot}}(t_0) = \mathbf{0}, \ddot{x}_{\text{foot}}(t_f) = \mathbf{0}$$

上述 7 个约束可以构成一个 6 项多项式

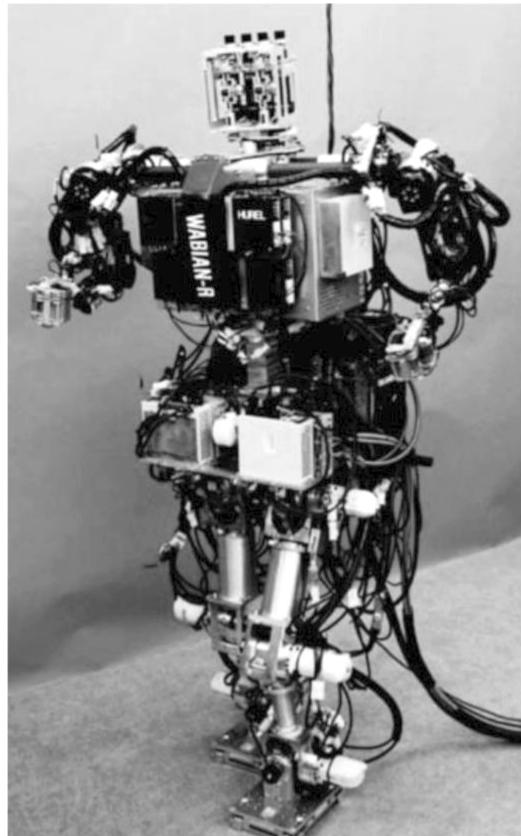


图 6-7 WABIAN-RII 机器人

$$\mathbf{x}_{\text{foot}}(t) = \mathbf{a}_0 + \mathbf{a}_1 t + \mathbf{a}_2 t^2 + \mathbf{a}_3 t^3 + \mathbf{a}_4 t^4 + \mathbf{a}_5 t^5 + \mathbf{a}_6 t^6$$

我们可以清楚地得到沿着路径的速度及加速度

$$\dot{\mathbf{x}}_{\text{foot}}(t) = \mathbf{a}_1 + 2\mathbf{a}_2 t + 3\mathbf{a}_3 t^2 + 4\mathbf{a}_4 t^3 + 5\mathbf{a}_5 t^4 + 6\mathbf{a}_6 t^5$$

$$\ddot{\mathbf{x}}_{\text{foot}}(t) = 2\mathbf{a}_2 + 6\mathbf{a}_3 t + 12\mathbf{a}_4 t^2 + 20\mathbf{a}_5 t^3 + 30\mathbf{a}_6 t^4$$

结合上述公式可以得到 7 个系数 (a_0, \dots, a_6) 的值。然后在 6 个多项式中我们可以得到行走的路线图。

(2) 快乐状态下的行走模式 如下表，在快乐状态下，我们设定 8 个参数和 4 个步态特征。

| 步态特征 | | | | |
|----------|-----------|-------|-------|----|
| 参数 | a | b | c | d |
| A: 步长 | 长 | 正常 | 正常 | 正常 |
| B: 脚 | M 形 | M 形 | 高 | 正常 |
| C: 脚趾 | 向外 | 直线 | 直线 | 直线 |
| D: 腰部 | 上下移动 | 正常 | 正常 | 正常 |
| E: 腰部 | 在滚动轴上旋转 | 固定 | 固定 | 固定 |
| F: 躯干和头部 | 大幅度前后摇摆躯干 | 只摇摆躯干 | 只摇摆脖子 | 固定 |
| G: 躯干和头部 | 躯干侧面摇晃 | 只摇摆躯干 | 只摇摆脖子 | 固定 |
| H: 胳膊 | 大幅度摆臂 | 正常 | 正常 | 正常 |

其中双足部分：

A：正常步伐的脚趾的中间 X 位置为 0.075m，最终 X 位置为 0.15m，而步长长的脚趾中间 X 位置为 0.10m，最终 X 位置为 0.20m。

B：正常脚在摆动阶段的第一个和最后一个阶段中，脚趾的加速度为 Z 方向的 0.002 m/s^2 。M 形脚踏轨迹在摆动阶段的第一个和最后一个阶段中，利用 0.05 m/s^2 到 Z 方向的加速度确定。在中心脚中，脚趾的 Y 方向位置是固定的。在步态阶段的中期，侧脚具有 -0.20 m 的趾部 Y 方向位置和 -20° 的侧倾定向。高脚在摆动阶段的中期具有 0.10 m 的趾部 Z 方向位置和 0.002 m/s^2 的 Z 方向加速度。

C：向外的脚具有 30.0° 的脚趾偏航方向。

(3) 难过状态下的行走模式 在难过状态下，我们设定 11 个参数和 2 个步态特征，如下表。

| 步态特征 | | |
|-------|---------|----|
| 参数 | a | b |
| A: 速度 | 慢 | 正常 |
| B: 头部 | 侧向倾斜 | 直立 |
| C: 脚趾 | 向里 | 直线 |
| D: 足部 | 拖着走 | 正常 |
| E: 躯干 | 从前到后摇摆 | 固定 |
| F: 躯干 | 向前弯曲 | 直立 |
| G: 躯干 | 向侧面弯曲 | 直立 |
| H: 躯干 | 在滚动轴上旋转 | 固定 |
| I: 步长 | 短 | 正常 |
| J: 头部 | 垂头 | 直立 |
| K: 胳膊 | 固定 | 摆动 |

其中双足部分：

A：正常速度为 1.28s/步，而慢速为 1.92s/步。

B：头部的初始侧倾角设置为侧向倾斜 20°。

C：脚趾的偏航角设置为 -10°，脚向内。

D：制作拖曳着行走的足部，在摆动的中间阶段，脚趾在 Z 方向位置为 0.02m。除此之外，在正常脚中脚趾在 Z 方向位置为 0.10m。

2. 变量分析

为了探索快乐和悲伤情绪下的双足行走，进行 3D 行走运动模拟。由模式生成器根据运动参数确定 32 个步行运动模式。软件使用 Windows 环境下的 Borland Builder C++ 和 OpenGL，可以记录 3D 情感行走的动画录像。然后，为了更逼真地展示出双足模型开始向前直行并且在最后一步绕偏转轴线向左转 90°，播放所记录约 15s 的 3D 动画用于评估。在这项研究中，早稻田大学的 17 位本科生作为评价者，分为 5 个级别评估了每个情感行走模式，5 个级别设为 0 点（最低点）~4 点（最高点）。每个快乐模式（左图）及难过模式（右图）的点的总和如图 6-8 所示。

| 模式 | 参数 | | | | | | | | 总和 | 参数 | | | | | | | | 总和 | | |
|----|----|---|---|---|---|---|---|---|----|----|---|---|---|---|---|---|---|----|---|----|
| | A | B | C | D | E | F | G | H | | A | B | C | D | E | F | G | H | I | K | |
| 1 | a | a | a | a | a | a | a | a | 33 | 1 | a | a | a | a | a | a | a | a | a | 31 |
| 2 | b | a | b | b | b | a | b | b | 6 | 2 | a | b | b | b | a | a | a | b | b | 18 |
| 3 | a | a | a | a | b | b | c | b | 22 | 3 | a | a | a | a | a | a | b | b | b | 25 |
| 4 | b | a | b | b | a | b | d | a | 11 | 4 | a | b | b | b | a | a | b | a | b | 28 |
| 5 | b | a | a | b | a | c | c | a | 10 | 5 | a | b | b | a | a | b | a | b | a | 20 |
| 6 | a | a | b | a | b | c | d | b | 35 | 6 | a | a | a | b | a | b | b | b | b | 16 |
| 7 | b | a | a | b | b | d | a | b | 5 | 7 | a | b | b | a | a | b | b | a | b | 6 |
| 8 | a | a | b | a | a | d | b | a | 37 | 8 | a | a | a | b | a | b | b | a | a | 35 |
| 9 | a | b | b | b | a | a | c | b | 4 | 9 | a | a | b | b | b | a | a | b | a | 24 |
| 10 | b | a | a | b | a | b | d | a | 24 | 10 | a | b | a | a | b | a | b | a | b | 26 |
| 11 | a | b | b | b | b | b | a | a | 16 | 11 | a | a | b | b | b | a | b | a | b | 31 |
| 12 | b | b | a | a | b | b | b | b | 12 | 12 | a | b | a | b | a | b | a | b | a | 37 |
| 13 | b | b | b | a | a | c | a | b | 24 | 13 | a | b | a | b | b | b | a | b | a | 13 |
| 14 | a | b | a | b | b | c | b | a | 12 | 14 | a | a | b | a | b | b | a | a | b | 16 |
| 15 | b | b | b | a | b | d | b | a | 45 | 15 | a | b | a | b | b | b | b | a | b | 8 |
| 16 | a | b | a | b | a | d | d | b | 5 | 16 | a | a | b | a | b | b | b | b | a | 34 |
| 17 | b | c | b | a | b | a | c | a | 37 | 17 | b | b | a | b | a | a | a | b | b | 22 |
| 18 | a | c | a | b | a | a | d | b | 4 | 18 | b | a | b | a | a | a | b | a | b | 27 |
| 19 | b | c | b | a | a | b | a | b | 25 | 19 | b | b | a | b | a | a | b | a | a | 24 |
| 20 | a | c | a | b | b | b | b | a | 13 | 20 | b | a | b | a | a | a | b | b | b | 24 |
| 21 | a | c | b | b | b | c | a | a | 18 | 21 | b | a | b | a | b | a | b | a | b | 14 |
| 22 | b | c | a | a | a | c | b | b | 29 | 22 | b | b | a | a | a | b | a | a | a | 23 |
| 23 | a | c | b | b | a | d | c | b | 13 | 23 | b | a | b | b | a | b | b | a | a | 18 |
| 24 | b | c | a | a | b | d | d | a | 24 | 24 | b | b | a | a | a | b | b | b | b | 18 |
| 25 | b | d | a | b | b | a | a | b | 6 | 25 | b | b | b | a | b | a | a | b | a | 15 |
| 26 | a | d | b | a | a | a | b | a | 30 | 26 | b | a | a | b | b | a | a | a | b | 28 |
| 27 | b | d | a | b | a | b | c | a | 9 | 27 | b | b | a | b | b | a | b | a | a | 39 |
| 28 | a | d | b | a | a | b | c | a | 10 | 28 | b | a | b | a | b | a | b | b | b | 27 |
| 29 | a | d | a | a | b | c | c | b | 36 | 29 | b | a | a | b | b | a | b | b | a | 28 |
| 30 | b | d | b | b | a | c | d | a | 9 | 30 | b | b | b | b | b | a | a | a | b | 12 |
| 31 | a | d | a | a | a | d | a | a | 37 | 31 | b | a | a | a | b | b | b | a | a | 29 |
| 32 | b | d | b | b | b | d | b | b | 2 | 32 | b | b | b | b | b | b | a | b | b | 7 |

图 6-8 快乐模式和难过模式的评估点

为了获得有效的情感行走模式，我们可以分析运动参数的方差值。其中，一组 n 个测量（情绪模式的数量）中的可变性与 SS（偏差平方和）成比例。

$$SS = \text{所有 } y \text{ 值的平方和} - CM$$

$$= \sum_{i=1}^p \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2 = \sum_{i=1}^p \sum_{j=1}^{n_i} y_{ij}^2 - CM \\ = SST + SSE$$

式中, y_{ij} 和 \bar{y} 分别表示 i, j 测量样本的方差和样本的平均值; CM 和 SST 分别表示平方值和平方和的校正; SSE 表示误差的平方和。式中, T_i 是处理值的总数。

$$\begin{aligned} \text{CM} &= \frac{\text{观测值总和}}{n} = \frac{\left(\sum_{i=1}^p \sum_{j=1}^{n_i} y_{ij} \right)^2}{n} = n \bar{y}^2 \\ \text{SST} &= \sum_{i=1}^p n_i (\bar{T}_i - \bar{y})^2 = \sum_{i=1}^p \frac{T_i^2}{n_i} - \text{CM} \\ \text{SSE} &= \text{SS} - \text{SST} \end{aligned}$$

参 考 文 献

- [1] H O Lim, A Ishii, A Takanishi. Basic emotional walking using a biped humanoid robot [C]. IEEE Int. Conf. Systems, Man, and Cybernetics, Tokyo, Japan, 1999: IV954-IV959.
- [2] A Takanishi, M Ishida, Y Yamazaki. The realization of dynamic walking by the biped walking robot [C]. IEEE Int. Conf. Robotics and Automation, St. Louis, MO, 1985: 459-466.
- [3] 王为. 基于情感计算的机器人学习系统研究 [D]. 浙江工业大学, 2009.
- [4] 王志良, 解仑. 我国人工心理与人工情感研究现状与进展 [J]. 中国科学基金, 2013 (1): 14-17.
- [5] 王志良. 人工心理与人工情感 [J]. 智能系统学报, 2006, 1 (1): 38-43.
- [6] Breazeal, Cynthia. *Designing Sociable Robots* [M]. The MIT Press, 2002.
- [7] Peter Menzel, Faith D'Aluisio. *Robosapiens* [M]. Cambridge: The MIT Press, 2000.
- [8] Mataric M J. Getting Humanoids to Move and Imitate [M]. IEEE Educational Activities Department, 2000.
- [9] Hu X, Xie L, Liu X, et al. Emotion Expression of Robot with Personality [J]. Mathematical Problems in Engineering, 2013 (5): 831-842.

第 7 章 双足步行机器人技术的课程实验

双足步行机器人是一个集多学科于一体的综合平台，制作双足步行机器人具有很强的实践性，只有读者亲自动手做多学科的相关实验，才能够更加深入地掌握其相关的制作技术。本章通过 PIC 单片机编程、舵机控制、机械结构设计与认识、动作调试、上位机编程、ZMP 实验、多智能体协调控制、语音控制等 12 个实验，讲授了双足步行机器人具体的制作方法与基本技术。

实验一 典型 PIC 单片机控制芯片的编程实验

一、实验目的

- 1) 熟悉 PIC 单片机和 MPLAB-IDE 编程环境。
- 2) 了解 MPLAB ICD2 在线调试器的作用及安装，熟悉连接和配置 MPLAB ICD2。
- 3) 学会创建工程文件、编写源文件代码并对源文件进行编译。
- 4) 熟悉使用 MPLAB ICD2 在线调试器的调试和下载程序。

二、实验设备

PC 一台、MPLAB 安装软件、MPLAB ICD2 在线调试器、目标电路板。

三、实验原理与说明

1. PIC 单片机简介

双足步行机器人的控制芯片为 Microchip 公司生产的 PIC 系列 16F877A 单片机，所以下面主要介绍的是 PIC 中档系列单片机 16F877A，16F877A 引脚排列如图 7-1 所示。

PIC 共有 35 条指令，每条指令字长为 14 位，其指令向上兼容，整个指令系统的指令按操作方式大体可分成 3 类：

- 1) 字节操作类指令。
- 2) 位操作类指令。
- 3) 立即数操作和控制操作类。

PIC 系列单片机是采用精简指令集计算（RISC）结构体系，所以寻址方式既少又简单。寻址方式根据操作数来源不同，可分为立即数寻址、寄存器间接寻址、直接寻址和位寻址 4 种。每条指令字长为 14 位，由于篇幅有限，仅在表 7-1 中列出了设计时所用到的字节操作指令、位操作指令和立即数操作与控制操作指令。表 7-2 是表 7-1 操作码字符的说明。

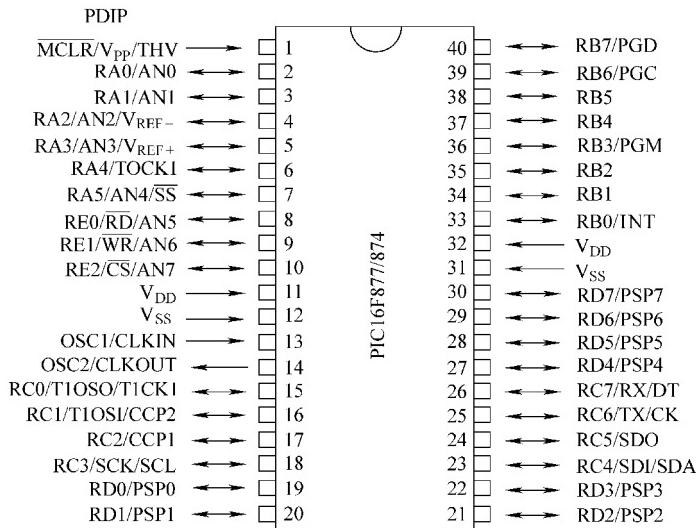


图 7-1 PIC16F877A 引脚排列

表 7-1 PIC16F877A 指令设置

| 助记符, 操作数 | 说 明 | 指令周期 | 影响的状态位 |
|-------------|--------------------|-------|--------|
| 字节操作类指令 | | | |
| ADDWF f, d | 将 W 和 f 相加 | 1 | C、DC、Z |
| ANDWF f, d | W 和 f ‘与’ 操作 | 1 | Z |
| CLRF f | 对 f 清零 | 1 | Z |
| CLRW - | 对 W 清零 | 1 | Z |
| DECFSZ f, d | f 减 1 操作, 若为 0, 跳转 | 1 (2) | Z |
| INCf f, d | f 加 1 操作 | 1 | Z |
| INCFSZ f, d | f 加 1 操作, 若为 0, 跳转 | 1 (2) | Z |
| IORWF f, d | W 与 f 同或 | 1 | Z |
| MOVF f, d | 传送 f | 1 | Z |
| MOVWF f | 将 W 送至 f | 1 | |
| NOP | 空操作 | 1 | |
| RLF f, d | f 寄存器带进位左循环 | 1 | C |
| RRF f, d | f 寄存器带进位右循环 | 1 | C |
| XORWF f, d | W 与 f 异或 | 1 | Z |
| 位操作类指令 | | | |
| BCF f, b | 清 f 的位 b | 1 | |
| BSF f, b | 置 f 的位 b | 1 | |
| BTFS f, b | 测试 f 的位 b, 为 0, 跳转 | 1 (2) | |
| BTFS f, b | 测试 f 的位 b, 为 1, 跳转 | 1 (2) | |
| 立即数与控制操作类指令 | | | |
| ADDLW k | 立即数加 W | 1 | C、DC、Z |
| ANDLW k | 立即数与 W 相与 | 1 | Z |
| CALL k | 调用子程序 | 2 | |
| GOTO k | 跳转 | 2 | |
| IORLW k | 立即数与 W 同或 | 1 | Z |
| MOVLW k | 立即数移到 W | 1 | |
| RETFIE - | 中断返回 | 2 | |
| RETLW k | 立即数送 W 子程序返回 | 2 | |
| RETURN - | 从子程序返回 | 2 | |
| XORLW k | 立即数与 W 相异或 | 1 | Z |

注：Z、DC、C 分别为状态 (Status) 寄存器中的零标志位、辅助进位/借位位、进位/借位位。

表 7-2 操作码字符说明

| 字符 | 说 明 |
|----|--|
| f | 寄存器地址 (0x00 ~ 07F) |
| d | 表示目的寄存器选择。d = 0，结果存于 W 中；d = 1，结果存于寄存器 f 中。默认值 d = 1 |
| W | 工作寄存器（累加器） |
| b | 某 8 位寄存器 f 内的位地址 |
| k | 立即数、常量或标号 |

以上都是汇编指令，不易记忆、不易读、不易移植。双足步行机器人采用通俗易懂的 C 语言来开发。

2. MPLAB-IDE 编程环境

MPLAB 集成开发环境（IDE）是综合的编辑器、项目管理器和设计平台，适用于使用 Microchip 公司的 PIC 系列单片机进行嵌入式设计的应用开发。

3. MPLAB ICD2 在线调试器

ICD2 是（in-circuit debugger）的英文缩写，中文是“在线调试”的意思。数字“2”，是因为它是 ICD1 的改进版，支持调试更多 Flash 存储器芯片。ICD2 的作用是让用户查看单片机内部的寄存器数据，帮助我们学习及发现程序问题。ICD2 本身既是一个调试器，同时也是一个烧入器。因为在线调试本身就是要求通过 ICD2 把目标代码下载到芯片内部，只不过调试时多写了一部分监控代码而已。如果不需要进行在线调试，就可以把 ICD2 看作为一个编程工具，把目标代码下载后，拔掉 ICD2 和目标板的连线，单片机就可以独立运行了。

四、实验步骤

1. 安装 PIC 单片机 MPLAB-IDE 编程环境及 ICD2 在线调试器并创建工程

(1) 安装 MPLAB IDE 从光盘安装，将光盘放入驱动器内，跟随屏幕上的菜单安装 MPLAB IDE。

也可浏览光盘，打开“softdown”文件夹，解压文件，双击其中的“MPLAB v7.50 Install.exe”进行安装。

MPLAB IDE 里没有自带 C 编译器，安装完 MPLAB IDE 后需要安装 Hi-tech PICC Compiler（参考安装说明书安装即可，此软件需自行下载或购买）。

(2) 安装 MPLAB ICD2 前面，我们已经安装完了 MPLAB IDE 编程环境，一般在安装目录下，有一个安装 MPLAB ICD2 的操作指南 ezicd2.htm：C:\Program Files\ MPLAB IDE\ Utilities\ MPUsbIRU\ ezicd2.htm，打开后按照它的说明一步一步地安装即可。如果目录下没有这个文档，则可按照与 MPLAB ICD2 配套的安装说明书进行安装即可。

(3) 运行 MPLAB IDE 双击桌面上 MPLAB IDE 的图标，或者选择 Start→Programs→Microchip MPLAB IDE→MPLAB IDE。屏幕上首先会显示 MPLAB IDE 的商标图案，然后出现 MPLAB IDE 的主窗口，如图 7-2 所示。

(4) 创建工程

1) 在图 7-2 所示的对话框中，选择菜单 Project→Project Wizard，单击“下一步”按钮，选择对话框中的 PIC16F877A 器件，在出现的如图 7-3 所示框中，单击“下一步”按钮，出现图 7-4 所示的对话框。

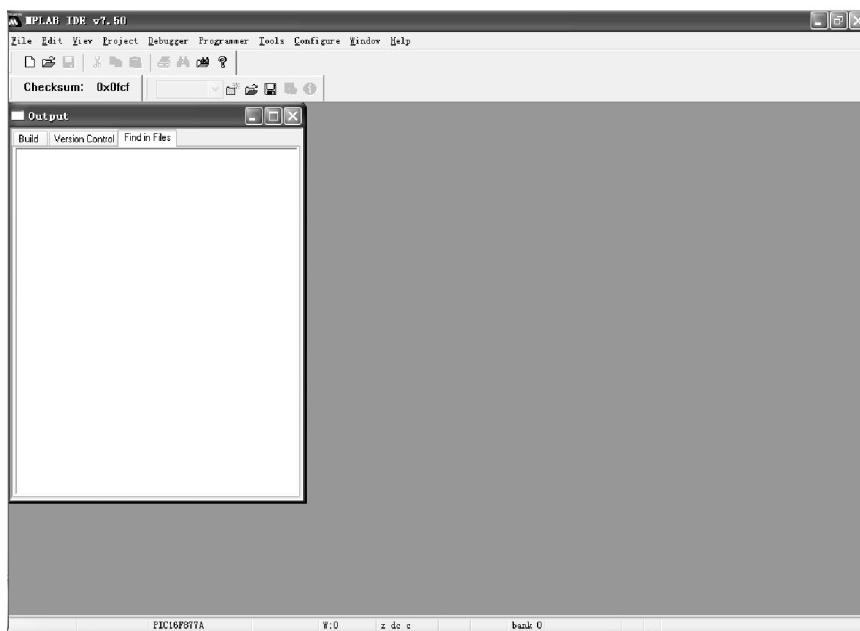


图 7-2 MPLAB IDE 的主窗口

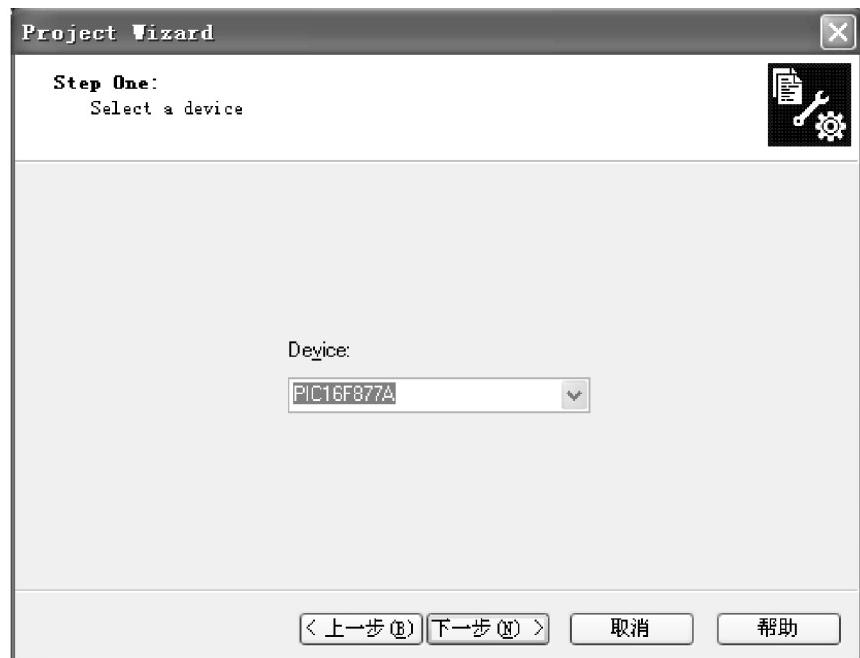


图 7-3 工程向导选择器件

- 2) 如图 7-4 所示的对话框中，选择 HI-TECH PICC Toolsuite 语言工具，单击“下一步”按钮，出现图 7-5 所示对话框。
- 3) 在图 7-5 所示对话框中，输入工程名和工程所在的路径，这里我们设工程名为 first，保存在 D:\frist 文件下，单击“下一步”按钮。出现图 7-6 所示的对话框。
- 4) 此时我们还没有哪些已存在的文件需要添加到工程中，因此直接点击下一步。

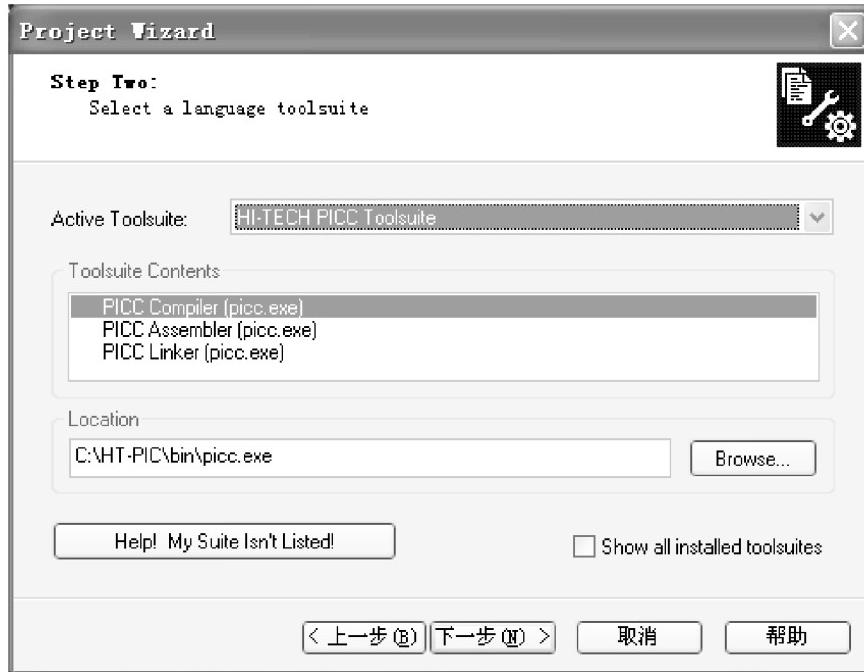


图 7-4 工程向导选择语言工具

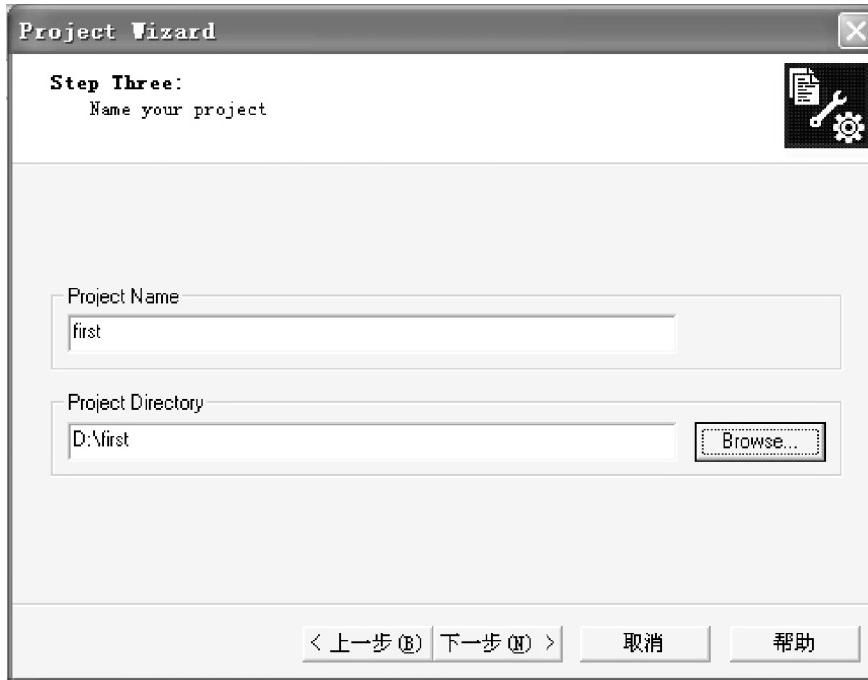


图 7-5 工程名和工程路径设置

- 5) 单击完成，项目工程建立完成。
2. 编译程序
 - 1) 单击 File→New 新建一个文件，编写程序代码，如下：

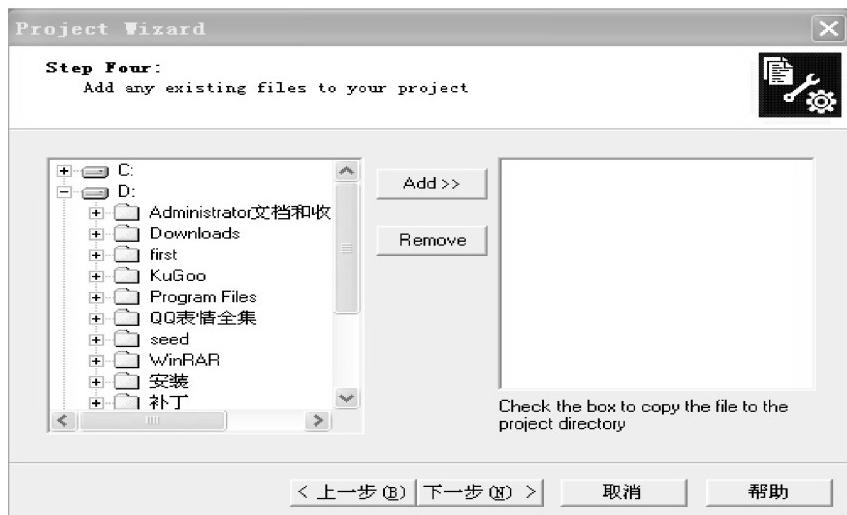


图 7-6 添加已存在的文件到工程中

```
#include <pic.h> //PIC16F877 的头文件
void main( void )
{
    unsigned int i = 0;
    for ( i;i < 10;i++ )
        printf( "This is my first PIC program" );
    while(1);
}
```

2) 单击 File→Save 保存这个文件到 D:\frist 文件下，文件名为 first.c。

3) 添加 first.c 到工程中去。

用右键单击 Source files，选择“Add Files”，找到 first.c，点击“打开”即可，具体操作如图 7-7、图 7-8 所示。

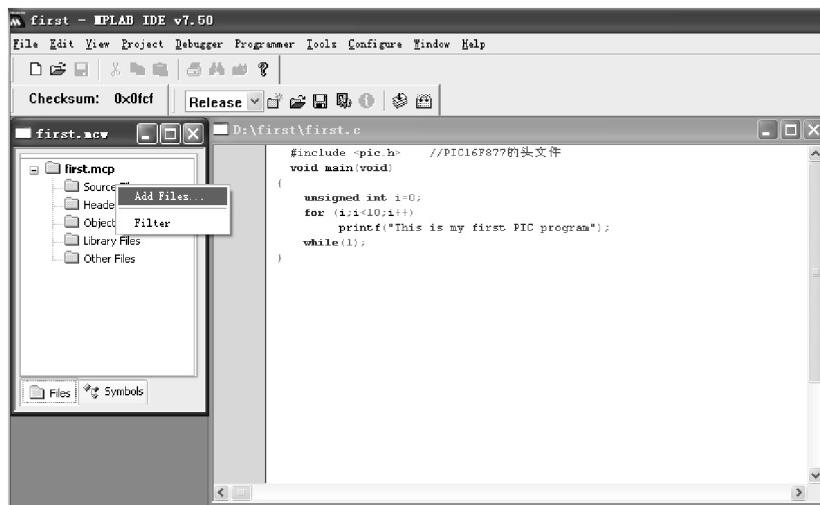


图 7-7 添加刚编译好的文件到工程中

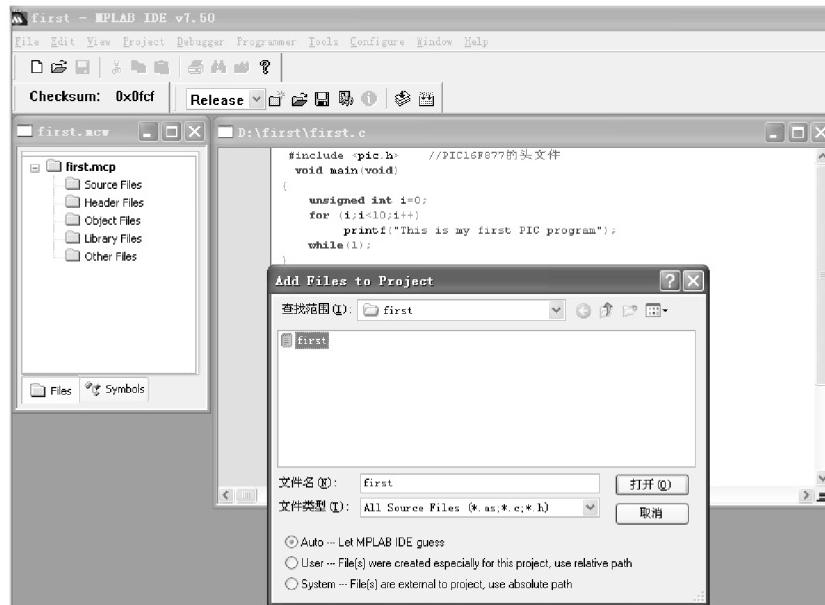


图 7-8 完成添加

- 4) 保存工程，单击 File→Save Workspace。
- 5) 单击 Project→Build All。在 Output 窗口中可以看到编译结果，如图 7-9 所示。

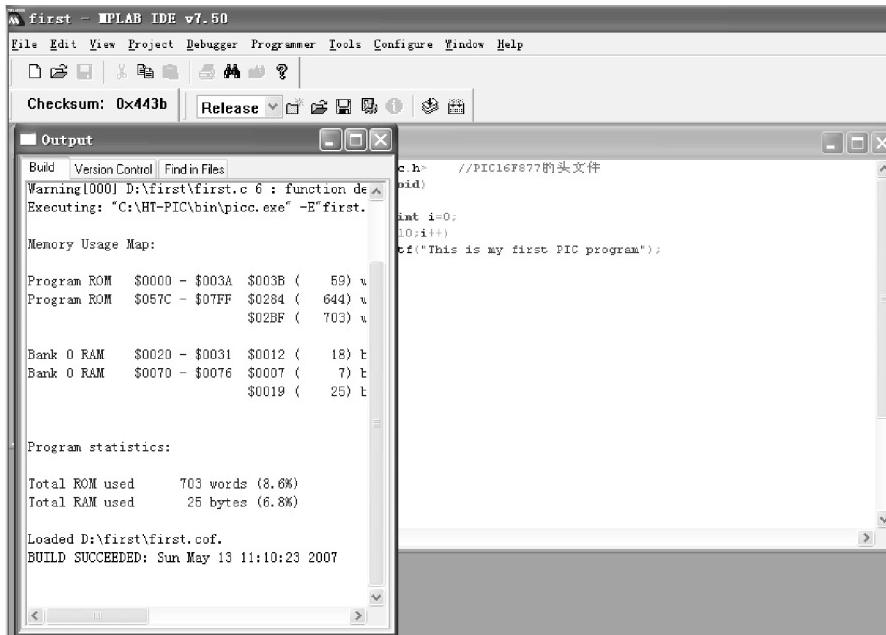


图 7-9 通过 Output 查看编译结果

假如文件汇编失败，检查下列两项，然后重新编译项目：

- ① 检查编辑器窗口中所输入代码的拼写和格式。如果汇编器在输出窗口中报告错误，则在该错误信息上双击鼠标，MAPLAB 会打开源代码窗口，并且在源代码窗口的左边空白处显

示绿色箭头标志。

②检查是否为 PICmicro 器件使用了正确的汇编器。选择 Project→Set Language Tool Locations，然后检查显示出的文件路径。如果路径正确，单击“Cancel”按钮；如果路径错误，则更改路径，然后单击 OK 按钮，如图 7-10 所示。

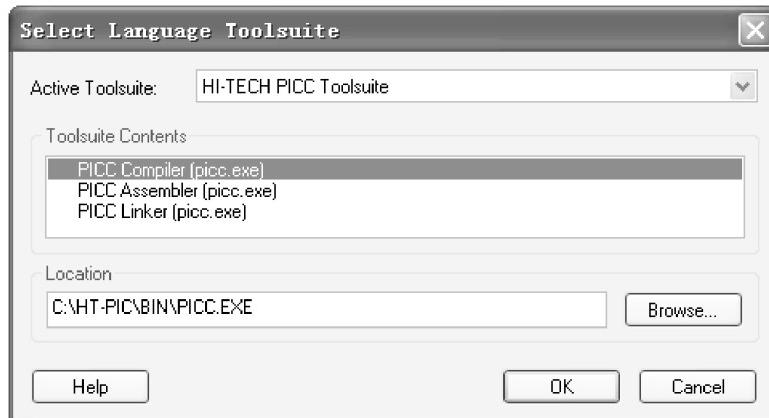


图 7-10 设置语言工具

6) 单击 View→Program Memory 可以观察程序存储器，如图 7-11 所示。

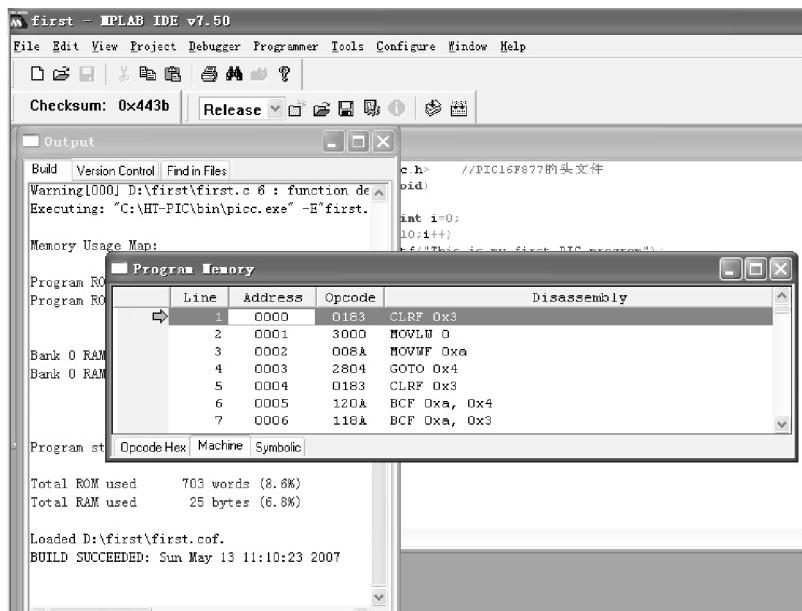


图 7-11 程序存储器

3. 连接 MPLAB ICD2

- 1) 将 ICD2 通过电缆连接到计算机上。
- 2) 用独立充电电池给目标演示板供电（为单向接插件）。
- 3) 将 ICD2 连接到目标演示板上（排线中的蓝色线靠近目标板电源）。

4) 选中 MPALAB IDE 的 Debugger→Select Tool→MPLAB ICD2，如图 7-12 所示。

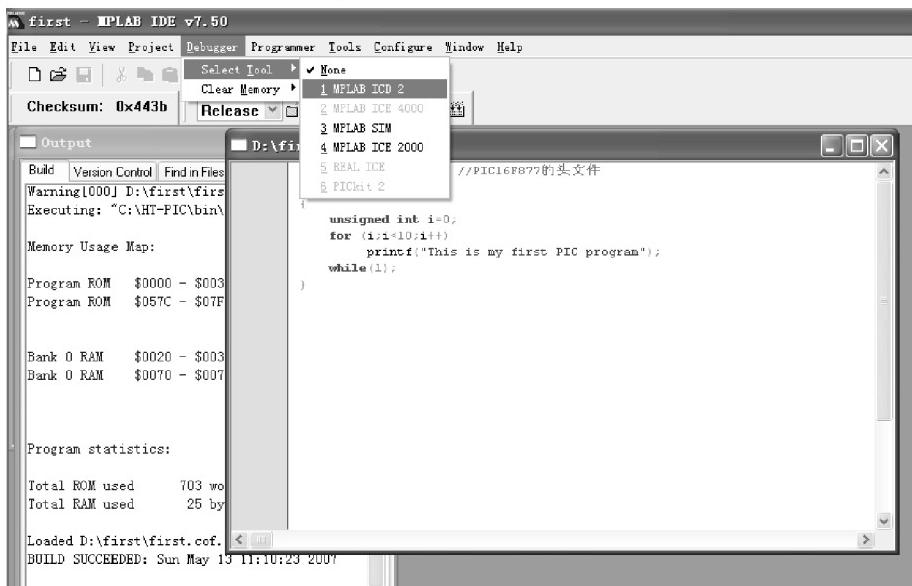


图 7-12 选择调试工具

5) 进入 MPLAB ICD2 Setup Wizard 一步一步设置，如图 7-13 所示。

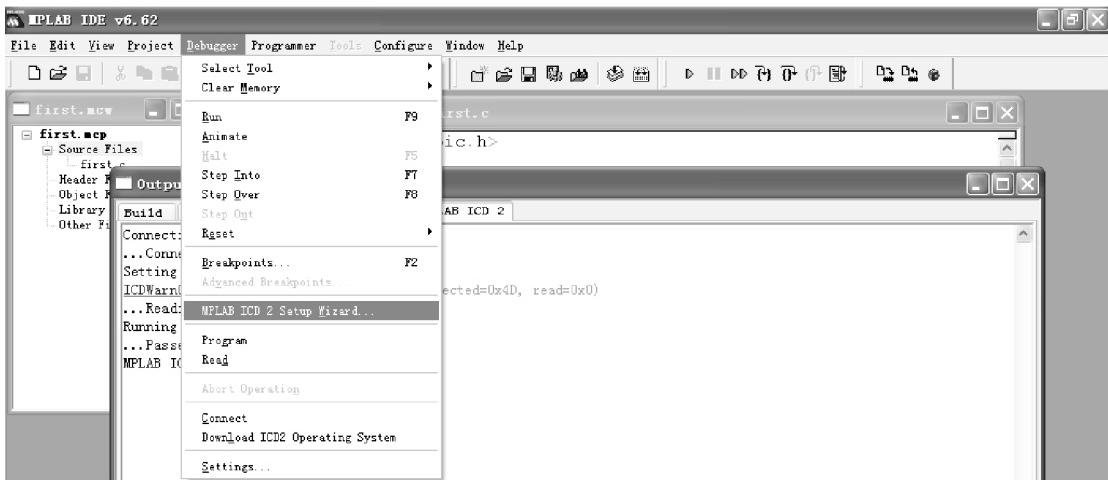


图 7-13 选择 ICD2 设置向导

步骤一：选择 ICD2 的 USB 连接模式，如图 7-14 所示。

步骤二：选择 ICD2 的供电方式，如图 7-15 所示。

步骤三：使能 ICD2 和 IDE 自动连接，如图 7-16 所示。

步骤四：使能 ICD2 自动下载所需的操作系统，如图 7-17 所示。

步骤五：单击“完成”按钮，完成向导设置，如图 7-18 所示。

6) 单击完成后，点击 Debugger→Connect，进行 ICD2 与目标板的通信连接，如图 7-19 所示。



图 7-14 选择 ICD2 的 USB 连接模式



图 7-15 设置 ICD2 的供电方式

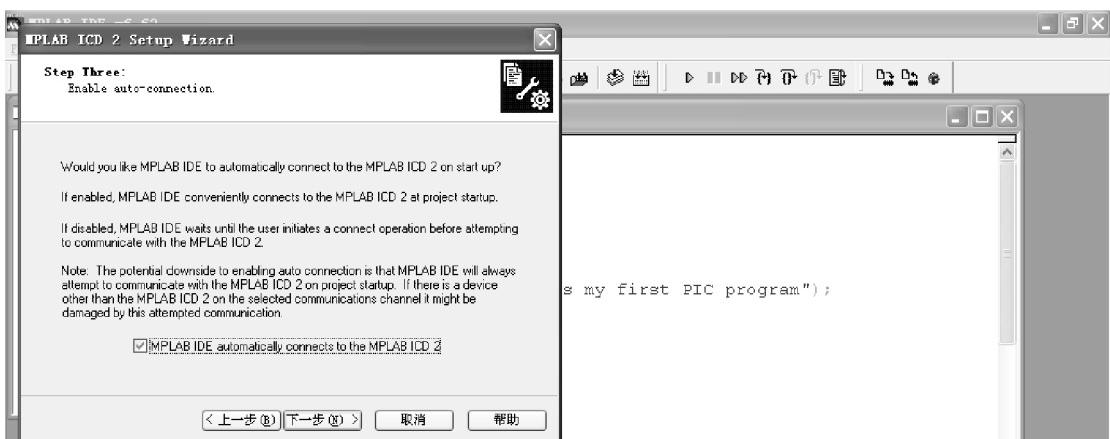


图 7-16 使能 ICD2 和 IDE 自动连接

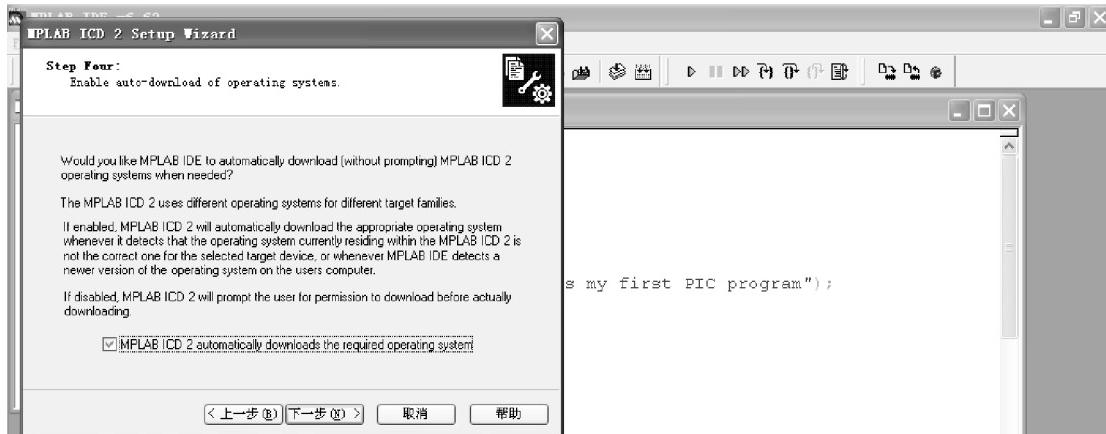


图 7-17 使能 ICD2 自动下载

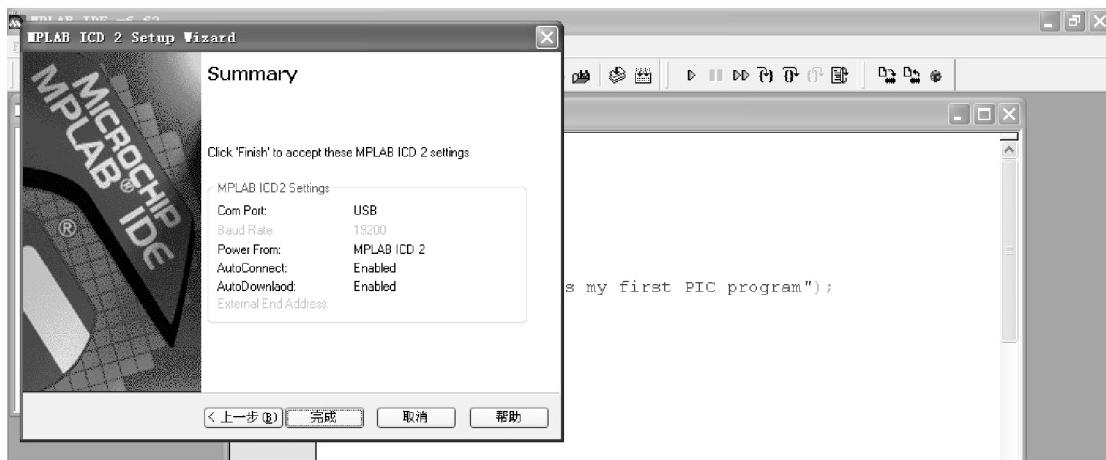


图 7-18 完成向导设置

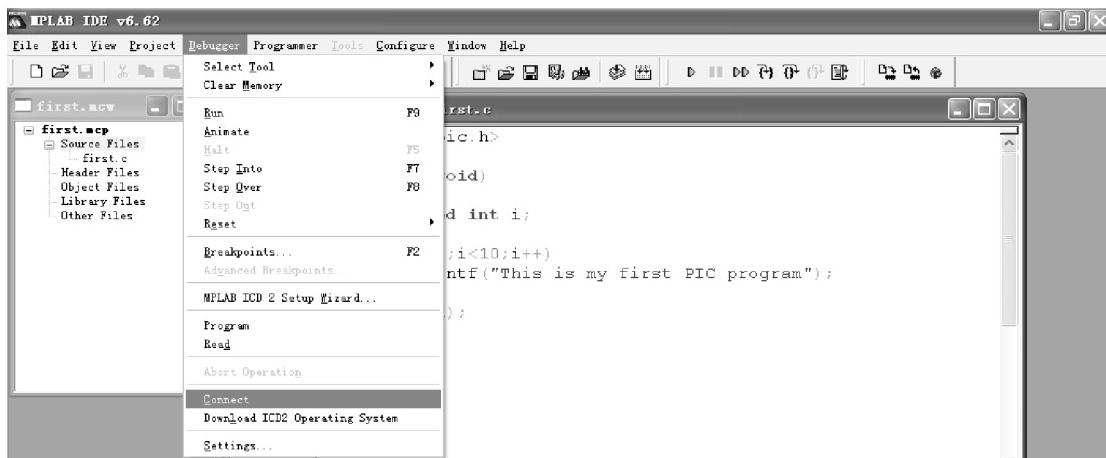


图 7-19 通过 ICD2 进行连接调试

如果出现了图 7-20 所示的信息，说明连接成功，可以用来调试了。

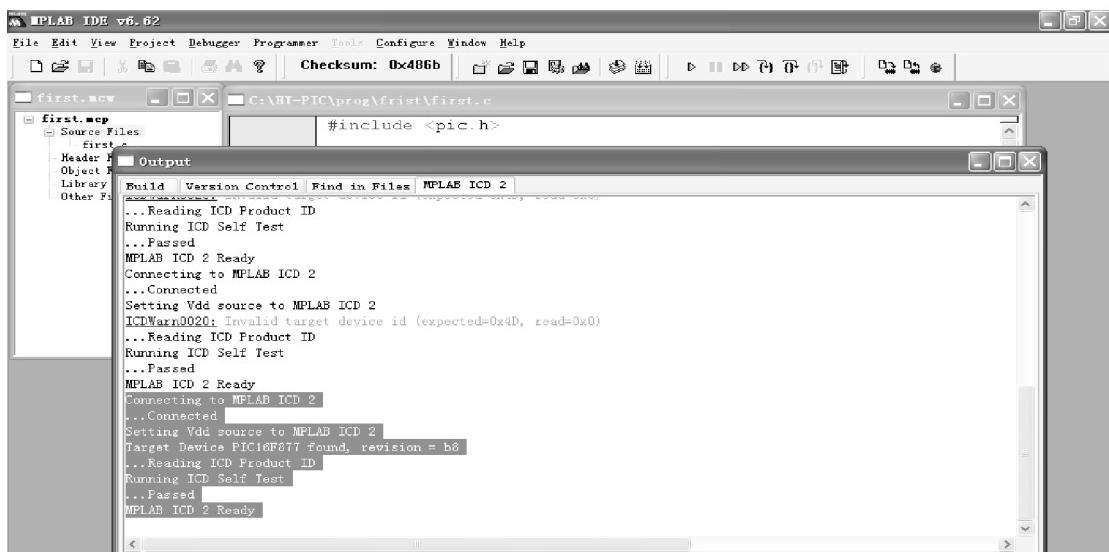


图 7-20 连接成功

4. 调试程序

1) 设置芯片的配置字，如图 7-21 所示，单击 Configure→Configuration Bits…，严格按照图 7-22 进行配置。

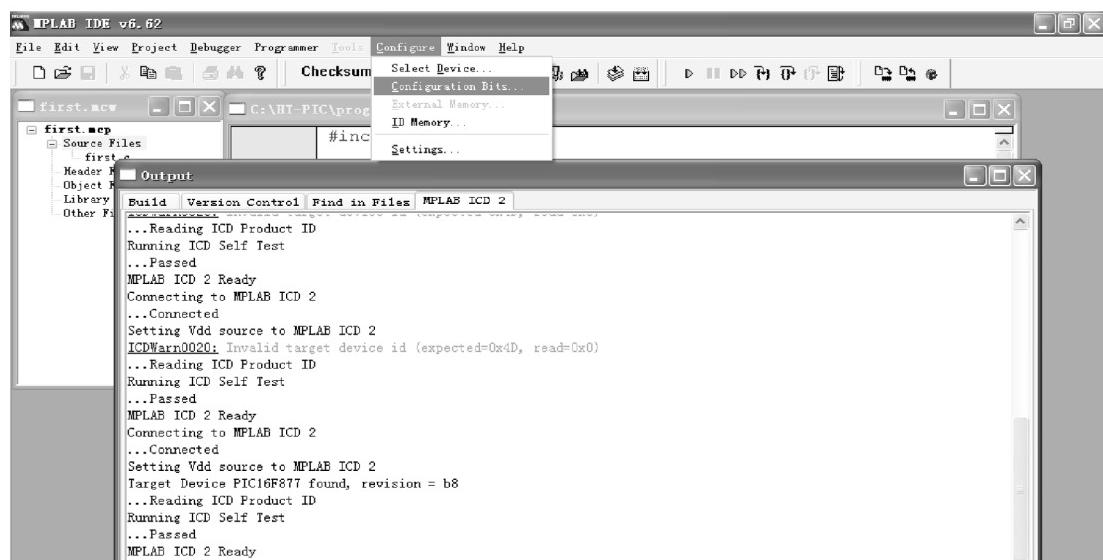


图 7-21 选择配置芯片配置字

2) 编译程序，单击 Project→Build All，如图 7-23 所示。

下下载程序，单击 Debugger→Program，如图 7-24 所示。

3) 下载完毕后，可以设置程序断点，观察程序执行情况及变量实时的值。首先单击 View→Watch，打开变量监视框，如图 7-25 所示。

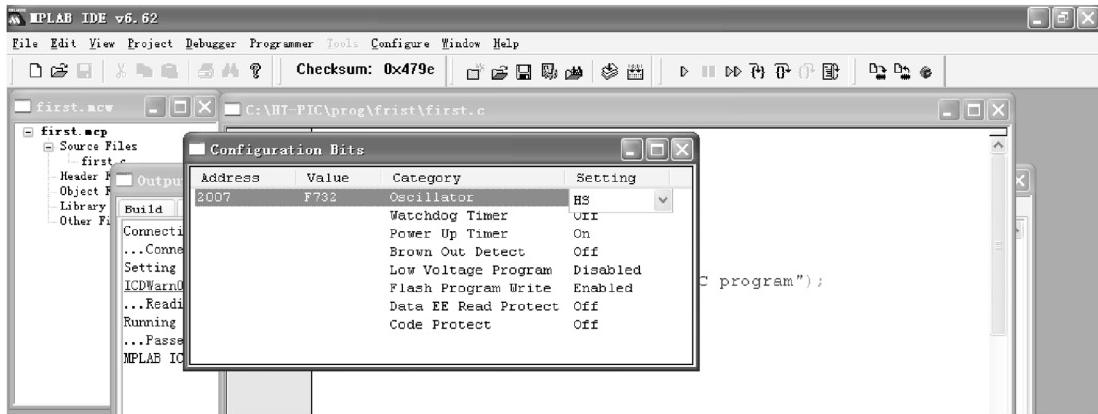


图 7-22 芯片配置字配置方法

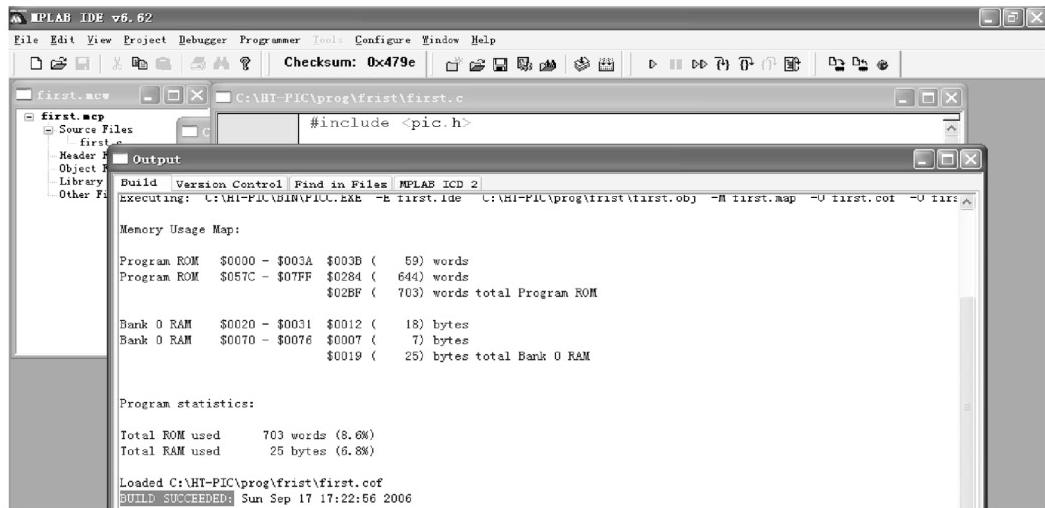


图 7-23 编译成功

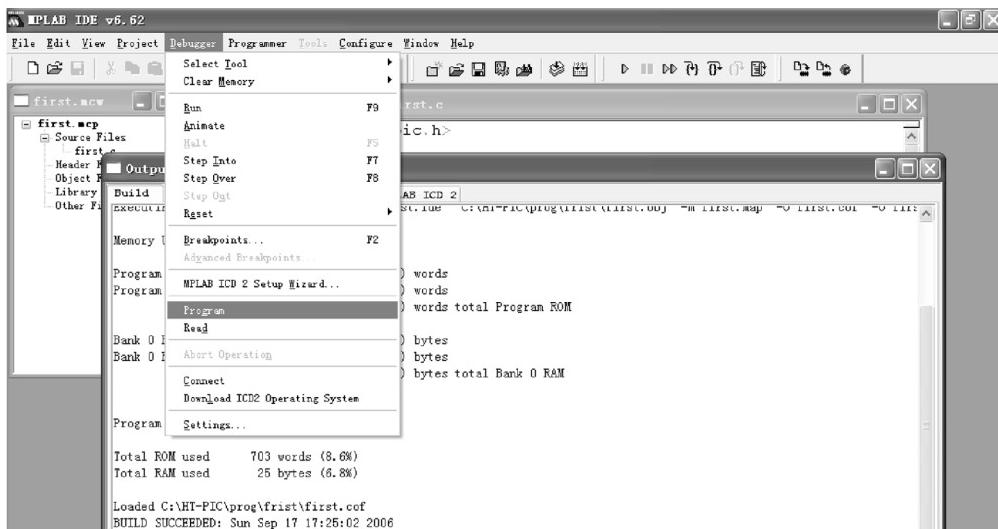


图 7-24 调试模式下进行程序下载

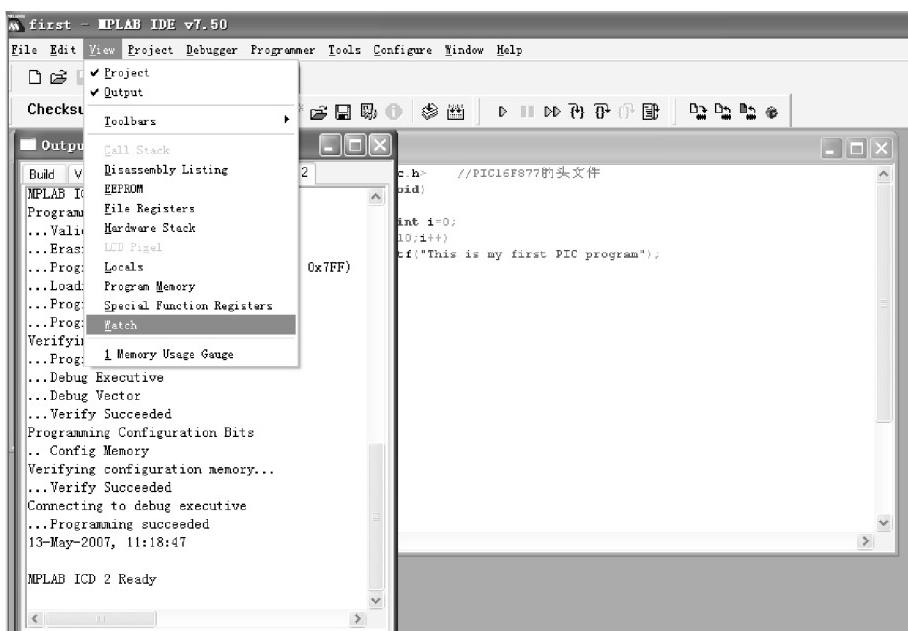


图 7-25 调出查看窗口对变量进行监视

比如我们要观察程序运行时变量 i 值的情况，单击“Watch”窗口的“Add Symbol”按钮，如图 7-26、图 7-27 所示。

设置断点、调试，可以全速运行、单步连续运行、单步跨过运行和复位四种模式来调试。如图中所示的语句处设置断点，则可双击该语句对应的左边灰色区域，则出现断点，然后再通过调试工具栏 中四种调试模式对应的按钮进行断点调试。以本实验为例，设置好断点后，不断点击 可以通过“Watch”窗口看到“i”值的变化情况，如图 7-28 所示。

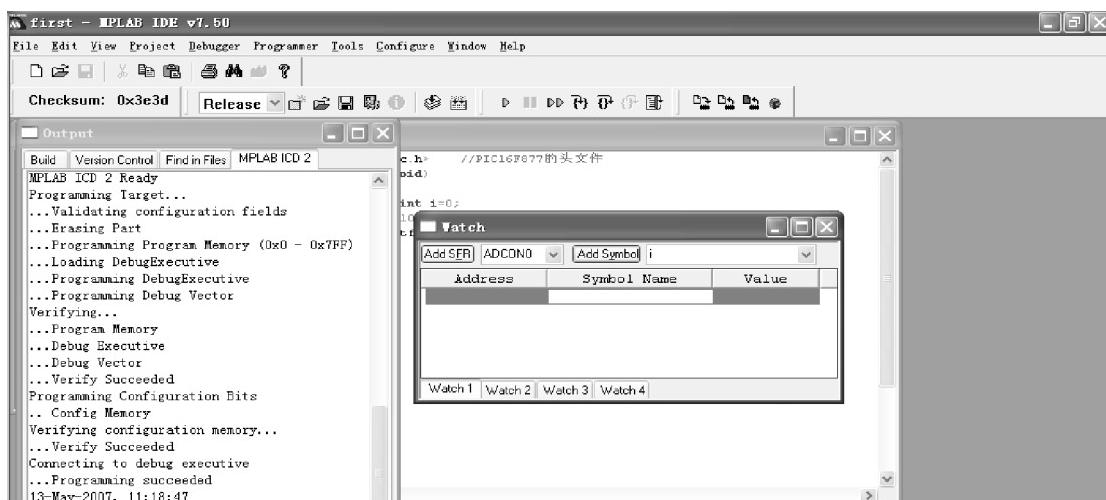


图 7-26 准备添加所要查看的变量到查看窗口

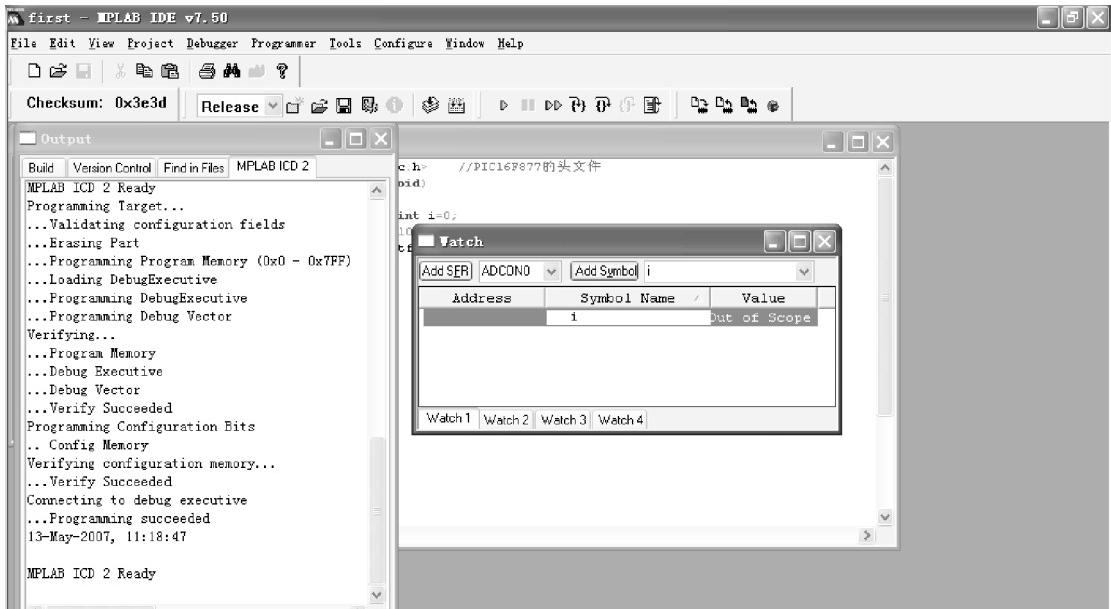


图 7-27 已添加所要查看的变量到查看窗口

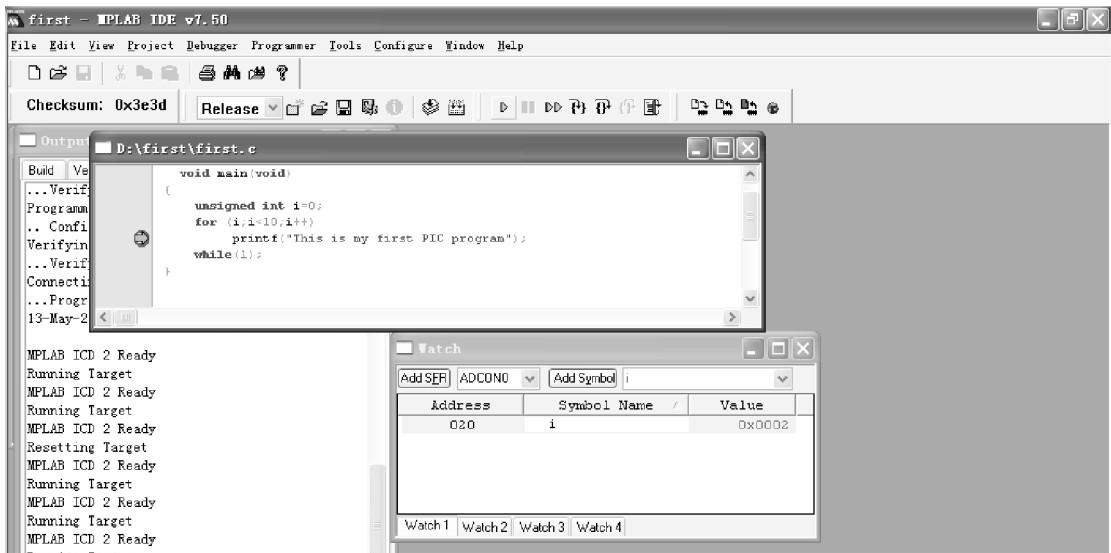


图 7-28 断点调试并查看变量值的变化

五、思考题

1. 创建工程的过程中应注意哪些问题?

答：1) 先要检查 MPLAB 软件是否正确安装完毕。

2) 检查 HT-PIC 软件是否正确安装完毕。

3) 在创建工程前确定本工程采用汇编语言还是 C 语言编写，从而确定是否选择 HI-TECH PICC Toolsuite 编译器。

4) 工程保存路径中不要包含中文字符。

2. 安装和配置需要注意哪些问题?

答: 1) 安装 MPLAB-ICD2 的驱动程序一定要按照说明文件安装, 以免出现不能连接等异常情况。

2) 仿真头的接线和单片机上的接头要配合正确, 否则容易烧坏仿真器。

3) 配置时要注意芯片的类型和设置芯片的配置字是否正确。

3. 调试模式和编程模式的切换需要注意哪些事项?

答: 1) 先关闭掉原来的模式, 比如当前是 Debug 模式, 先进 Debugger→SelectTools 下设成 None , 再进入 Programer 去选。

2) 切换后芯片的配置字会发生改变, 需要重新配置, 否则程序可能无法正确运行。

实验二 常用传感器实验

作为感知、采集、转换、传输和处理各种信息必不可少的功能器件传感器, 已成为与计算机同等重要的技术工具。本次实验主要让同学们了解超声波传感器、霍尔传感器、温度传感器、红外传感器。

一、实验目的

- 1) 熟悉机器人技术中常见传感器的原理和使用。
- 2) 综合使用 PIC 单片机资源, 理解中断定时器的原理。
- 3) 巩固 C 语言能力, 掌握传感器使用的时序关系。

二、实验设备

PC 一台 (MPLAB IDE 环境)、MPLAB ICD2 一台、实验板一块 (集成各种传感器)、供电电源 (12V)、万用表一台、示波器一台及数根接插线。

实验板载传感器: 超声波传感器 (接受探头、发送探头)、温度传感器 (DS18B20)、霍尔传感器、光电传感器。

三、实验原理与说明

1. 霍尔传感器

霍尔传感器有以下几个主要应用: 磁场测量、电流测量、转速和脉冲测量、信号的运算和测量、拉力和压力测量。本次试验采用的传感器是磁场测量传感器。它的最简单的原理就是当传感器连接到电路中以后, 当其处在强磁场中时它的 DQ 信号线就会输出低电平, 离开强磁场的时候它输出高电平。但是它对磁场的极性是有要求的, 必须用正确的极性对准霍尔传感器, 它才能输出低电平。我们利用这种特性可以应用在很多地方, 比如在机器人的导轨上适当地加上几个霍尔传感器, 通过这种方式我们就可以定位出导轨上的机构所在的位置了。它的外观及连线如图 7-29 所示。

2. 光电传感器

对射式光电传感器包含在结构上相互分离且光轴相对放置的发射器和接收器, 发射器发

出的光线直接进入接收器。当被检测物体经过发射器和接收器之间且阻断光线时，光电传感器就产生了开关信号。当检测物体是不透明时，对射式光电传感器是最可靠的检测模式。

这种遮挡式光传感器可以应用在很多地方，比如电动机转速的测定。安装在门的两侧可以用来防盗。

在对射管没有被不透明物体遮挡时，发射管发出的光线能被接收管接收到，这时对射管的信号线输出低电平，当发射管的光线被不透明的物体遮挡住时，接收管不能接收到信号，这时对射管输出高电平。如图 7-30 所示。

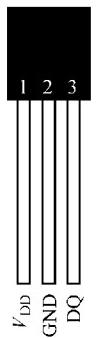


图 7-29 霍尔传感器外观及连线图

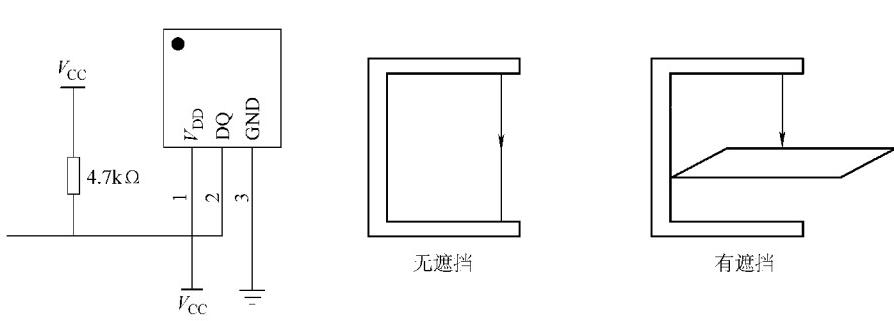


图 7-30 光电传感器发射接收端有无遮挡示意图

3. 温度传感器

新型数字温度传感器 DS18B20 具有体积更小、精度更高、适用电压更宽、采用一线总线、可组网等优点，在实际应用中取得了良好的测温效果。

DS18B20 的主要特性：

- 1) 适应电压范围更宽，电压范围为 3.0 ~ 5.5V，在寄生电源方式下可由数据线供电。
- 2) 独特的单线接口方式，DS18B20 在与微处理器连接时仅需要一条口线即可实现微处理器与 DS18B20 的双向通信。
- 3) DS18B20 支持多点组网功能，多个 DS18B20 可以并联在唯一的三线上，实现组网多点测温。
- 4) DS18B20 在使用中不需要任何外围元件，全部传感元件及转换电路集成在形如一只晶体管的集成电路内。
- 5) 温度范围 $-55 \sim +125^{\circ}\text{C}$ ，在 $-10 \sim +85^{\circ}\text{C}$ 时精度为 $\pm 0.5^{\circ}\text{C}$ 。
- 6) 可编程的分辨率为 9 ~ 12 位，对应的可分辨温度分别为 0.5°C 、 0.25°C 、 0.125°C 和 0.0625°C ，可实现高精度测温。
- 7) 在 9 位分辨率时，最多在 93.75ms 内把温度转换为数字，12 位分辨率时最多在 75.0ms 内把温度值转换为数字，速度更快。
- 8) 测量结果直接输出数字温度信号，以“一线总线”串行传送给 CPU，同时可传送 CRC 校验码，具有极强的抗干扰纠错能力。
- 9) 负压特性。电源极性接反时，芯片不会因发热而烧毁，但不能正常工作。它的外形及连线如图 7-31 所示。

4. 超声波传感器

超声测距是一种非接触式的检测方式。与其他方法相比，如电磁的或光学的方法，它不受光线、被测对象颜色等影响。对于被测物处于黑暗、有灰尘、烟雾、电磁干扰、有毒等恶劣的环境下有一定的适应能力。因此在液位测量、机械手控制、车辆自动导航、物体识别等方面有广泛应用。特别是应用于空气中的测距，由于空气中波速较慢，其回波信号中包含的沿传播方向上的结构信息很容易检测出来，具有很高的分辨力，因而其准确度也较其他方法为高；而且超声波传感器具有结构简单、体积小、信号处理可靠等特点。这种技术常被应用于避障技术中。同学们掌握这个传感器的使用方法，对学习机器人技术是非常有帮助的。

超声波测距的原理如图 7-32 所示。

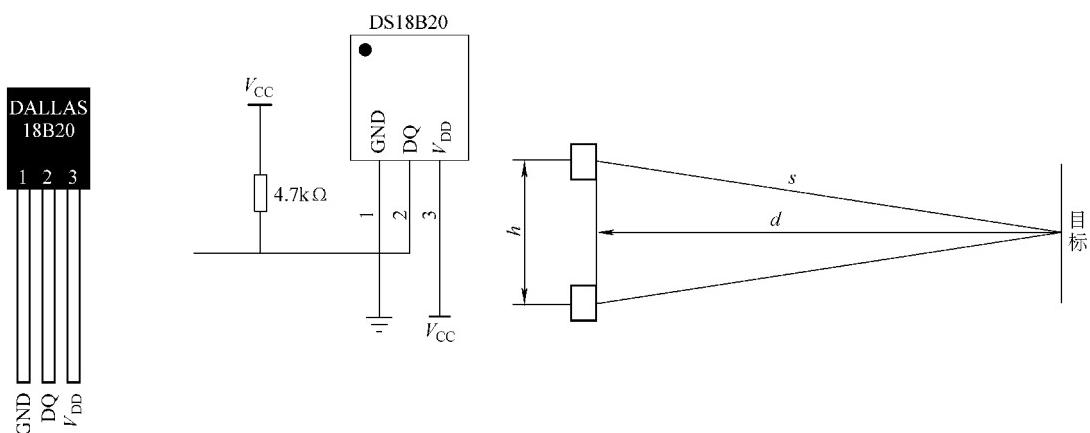


图 7-31 温度传感器 DS18B20 外观及连线图

图 7-32 超声波传感器测距原理图

发射探头发射超声波，当超声波遇到障碍物后声波被反射回来，并被接受探头接收到，控制单片机记录超声波从发射到接受超声波的时间，利用超声波在空气中的传播速度就可以计算出障碍物的距离： $s = vt/2$ 。

我们使用的超声波传感器是压电式超声波传感器 ZR40-16、ZT40-16，它们的原理如图 7-32 所示，压电式超声波发生器实际上是利用压电晶体的谐振来工作的。超声波发生器内部结构如图 7-33 所示，它有两个压电晶片和一个共振板。当它的两极外加脉冲信号，其频率等于压电晶片的固有振荡频率时，压电晶片将会发生共振，并带动共振板振动，便产生超声波。反之，如果两电极间未外加电压，当共振板接收到超声波时，将压迫压电晶片作振动，将机械能转换为电信号，这时它就成为超声波接收器了。

但是接收到的信号是非常微弱的，所以我们要把它放大，以使单片机能够检测到它的大小。

四、实验步骤

实验板的布置如图 7-34 所示。

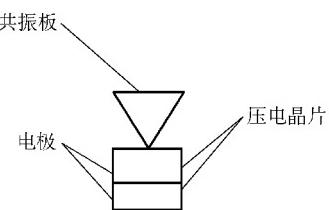


图 7-33 超声波传感器
内部结构图

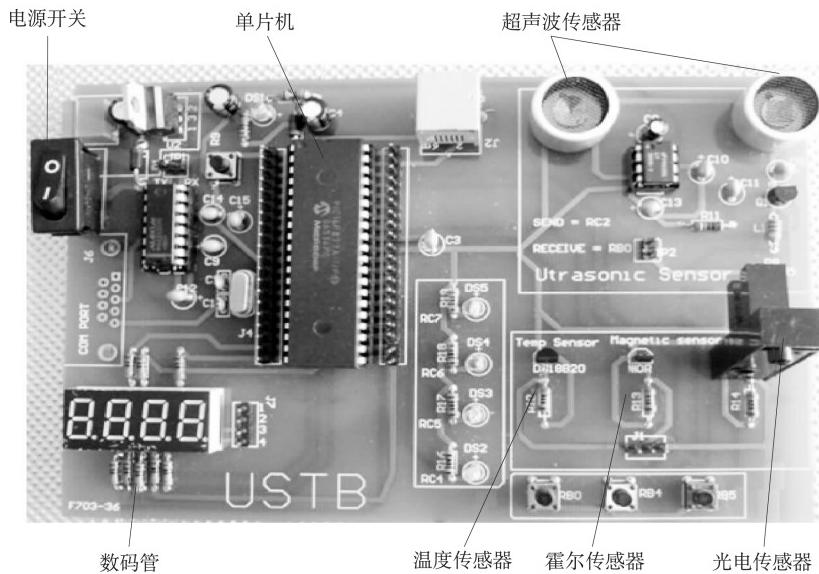


图 7-34 传感器实验板布置（正式）图

1. 霍尔传感器

要求：连接完整的电路，在单片机上下载程序实现在奇数次中断时（霍尔传感器在奇数次检测到磁场时）所有 LED 灯循环点亮，偶数次中断时（霍尔传感器在偶数次检测到磁场时）关闭 LED。

步骤：

1) 给试验板加电（注：电源电压 12V，电源极性不能接反，当电源部分指示灯点亮时表明试验板加电正确）。

2) 用万用表测量霍尔传感器的信号线，信号线为 J1 端子中间的插针，地线为 J5 端子的上测，观察电平值，这时将强磁铁靠近霍尔传感器，观察电压值是否变低了，如果电压值没有足够低，这时把磁铁翻过一面再靠近霍尔传感器，然后再观察霍尔传感器的信号线的电平变化，霍尔传感器的检测结果是否存在极性问题，请同学们在实验室验证。

3) 观察实验板原理图，找出单片机的 RB0（33 号引脚），并用插接线连接霍尔传感器的信号线与单片机的外部中断引脚——RB0。

4) 在 MPLAB IDE 环境下利用 MPLAB ICD2 把源程序下载到单片机中，步骤如实验一中所叙述，并用磁铁验证该实验。

霍尔传感器在奇数次检测到磁场时如图 7-35 所示，此时 4 个灯全亮。

霍尔传感器在奇数次检测到磁场时，此时 4 个灯全灭。请同学们做完每个实验后将接插线拆除。

5) 仔细阅读源代码，并说明源代码实现的功能（完成思考题 1）。

6) 请同学们试着独立编写程序完成上面要求的功能。

2. 光电传感器

要求：用单片机的中断功能记录光电对射管被遮挡的次数，并在四位 LED 数码管的低两位显示出来。只要求记录 99 次以下的遮挡次数。

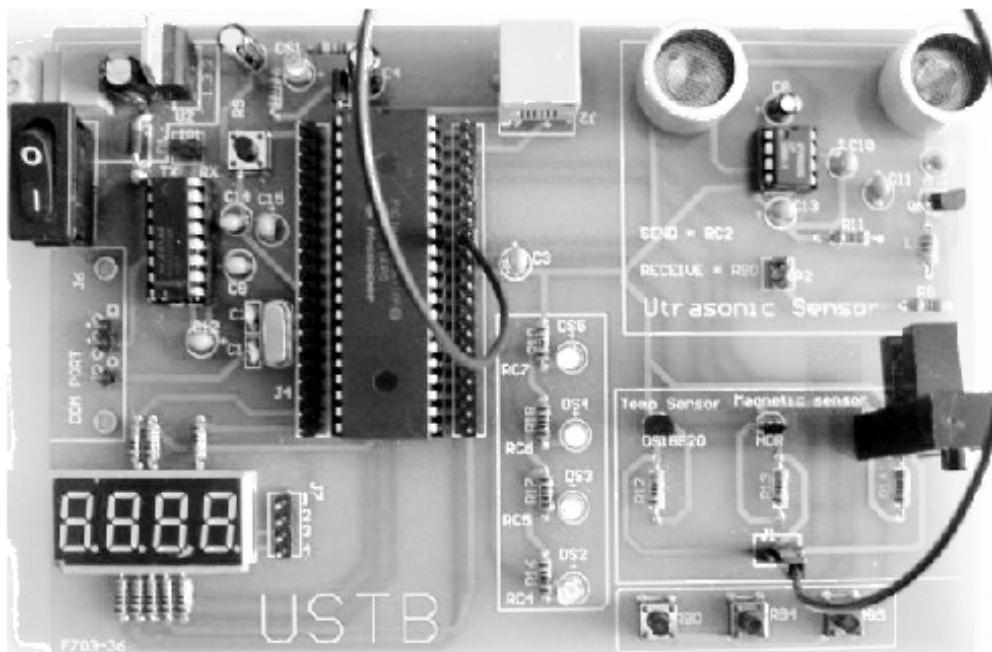


图 7-35 霍尔传感器实验演示图

步骤：

1) 用万用表测量光电传感器的信号线，并观察其电平值，这时用一个纸片挡在发射管与接受管之间，再观察电压值是否变低了，撤离遮挡物以后电平恢复原置值，注意光电传感器的信号线为 J1 端子右侧第 3 个插针。

具体操作如图 7-36 所示。

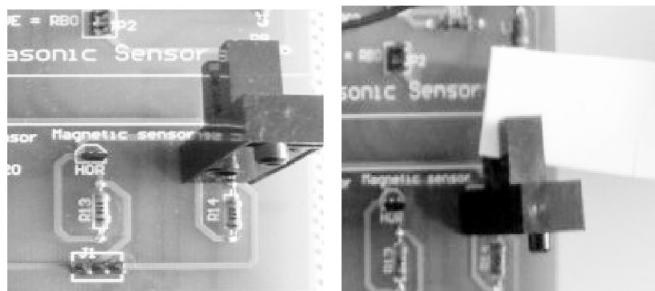


图 7-36 传感器操作示意图

2) 观察实验板原理图，找出单片机的 RB0 引脚，然后用插接线连接光电传感器的信号线与单片机的外部中断引脚——RB0，并对照程序与实验板原理图用接插线连接四位 LED 数码管与单片机，J7 端子插针由下向上分别为个、十、百、千，因此 J7 端子个位插针与单片机 RA3 (5 号引脚) 相连，J7 端子十位插针与单片机 RA2 (4 号引脚) 相连，其他不连，最后运行程序后两位数码管显示出遮挡次数。

- 3) 在 MPLAB IDE 环境下利用 MPLAB ICD2 把源程序下载到单片机中，并验证实验。
- 4) 仔细阅读参考源代码并理解每条语句的相关功能（完成思考题 2）。

5) 独立编写程序完成上述要求的功能。

3. 温度传感器

要求：使用温度传感器测定周围环境温度，用摄氏度为单位，并且通过串口通信发送到计算机的串口助手中显示出来，要求波特率为 9600bit/s，并且在数码管上显示温度值精确到摄氏度。

步骤：

1) 观察温度传感器的外形，这里温度传感器的信号线在 J1 端子左边第 1 个插针，同学们在以后的独立开发应用中不要弄错。

2) 观察实验板原理图，找出单片机的 RA1 (3 号引脚)，用插接线连接温度传感器的信号线与单片机的外部中断引脚——RA1 (可以连到其他普通 IO 口上，本程序以连接到 RA1 为例)，并对照程序用接插线连接四位 LED 数码管与单片机，J7 端子个位与单片机 RC1 (16 号引脚) 相连，J7 端子十位与单片机 RC0 (15 号引脚) 相连。

3) 在 MPLAB IDE 环境下利用 MPLAB ICD2 把源程序下载到单片机中，并在实验板上验证。

4) 仔细阅读温度传感器的源代码，在阅读的过程中，把主要精力放在主函数上，了解温度传感器的操作顺序 (命令的发送顺序)，各个功能模块函数已经实现，由于同学们的课程时间较紧张，在这里不要求同学们自己编写了，完成思考题 3 即可。

4. 超声波传感器

超声波的发送电路是通过 LC 振荡实现电压放大的，而在振荡的过程中存在瞬间短路现象，如图 7-37 所示，当 SEND 为高电平时，晶体管处于导通状态，这是电源通过电感经过晶体管直接流入地，这个瞬间要求很短，经过一个短暂的瞬态，电感中的电流已经达到了一定的大小，此时 SEND 变为低电平，晶体管截止，电流不能通过晶体管，但是由于电感的自感效应，会在电容量两端充电，并且电压升到将近 30V，如此反复，就在晶体管的集电极形成了振荡的电压，我们给出的 SEND 端信号为 40kHz 的方波信号，而在集电极处我们得到了峰-峰值约为 30V 的振荡电压，我们将这个电压加到超声波的发射探头上就可以发射一定功率的超声波了。

而这种发射电路，就决定了我们绝对不能让 SEND 端的高电平时间过长，只要不是在发射状态就必须要求 SEND 端处于低电平状态。

电路示意如图 7-37 所示。

要求：实现超声波测距，并且在数码管上显示出具体值。

步骤：

1) 观察实验板原理图，找出单片机的 RB0 (33 号引脚) 与 RC2 (17 号引脚)，用接插线将 SEND 端 (JP2 端子的下侧插针) 与 RC2 相连 (RC2 是单片机的 PWM 输出端)，将 RECEIVE 端 (JP2 端子的上侧插针) 与 RB0 相连 (外部中断)。注意：板子上的 SEND 和 RECEIVE 写反了。

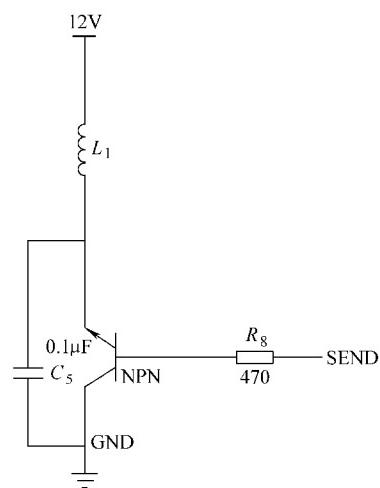


图 7-37 超声波传感器振荡电路示意

2) 对照程序与实验板原理图用接插线连接四位 LED 数码管与单片机, J7 端子个、十、百、千插针分别与单片机 RA3 (5 号引脚)、RA2 (4 号引脚)、RA1 (3 号引脚)、RA0 (2 号引脚) 相连。

3) 在 MPLAB IDE 环境下利用 MPLAB ICD2 把源程序下载到单片机中, 并对照程序与实验板原理图用接插线连接四位 LED 数码管与单片机。

超声波传感器实验运行结果如图 7-38 所示。

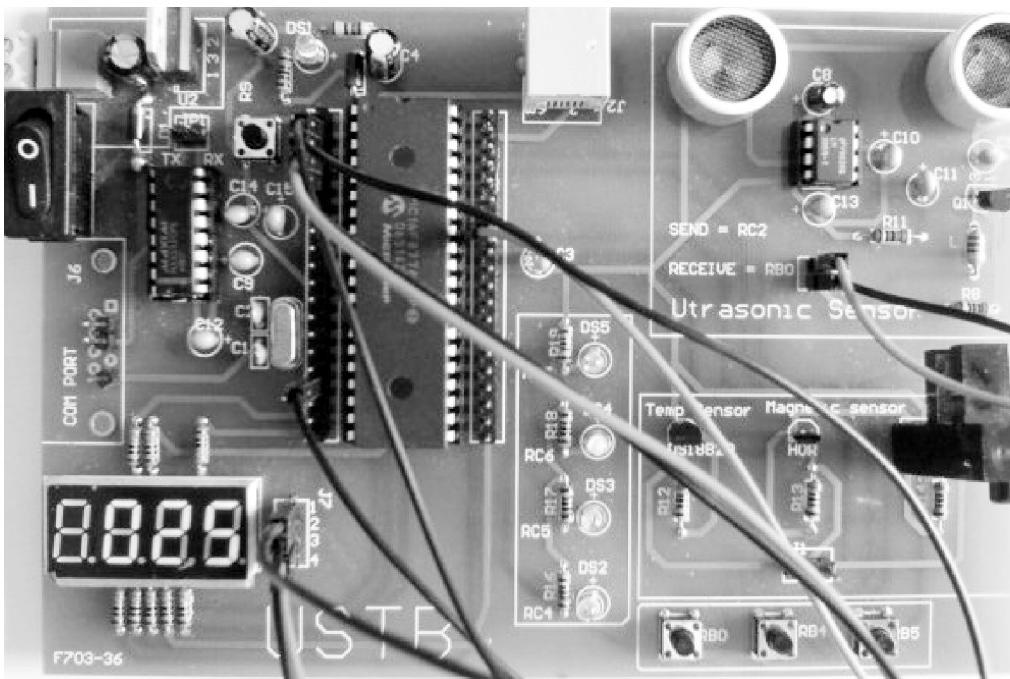


图 7-38 传感器实验运行结果

4) 用示波器观察发射探头上的波形, 认真体会超声波测距的原理。用示波器观察接收端的信号波形, 用两路示波器同时观察发送与接收端的波形, 体会时差测距的思想。

5) 仔细阅读给出的源代码并理解相关语句的功能 (完成思考题 4)。

五、思考题

1. 在霍尔传感器实验中给出的源代码实现的功能是什么? 用简洁的语言概括出来。

答: 在奇数次中断到来时点亮所有 4 个 LED 灯, 偶数次中断到来时关闭所有 LED 灯。

2. 设计一个光电对射管的简单应用。谈谈自己的创意。

利用光电开关还可以进行产品流水线的产量统计、对装配件是否到位及装配质量进行检测, 例如瓶盖是否压上、商标是否漏贴 (见图 7-39), 以及送料机构是否断料 (见图 7-40) 等。

此外, 利用光电传感器可以检测布料的有无和宽度。利用遮挡式光电传感器检测布料的下垂度, 其结果可用于调整布料在传送中的张力, 利用安装在框架上的光电传感器可以发现漏装产品的空箱, 比利用油缸将空箱推出。

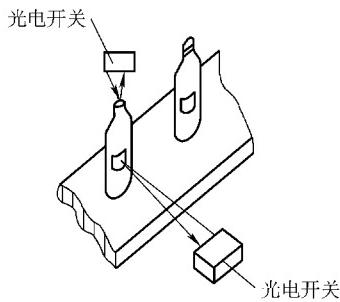


图 7-39 瓶子罐装检测示意图

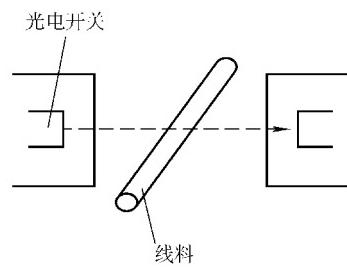


图 7-40 送料机构检测示意图

3. 总结温度传感器的命令发送的顺序?

答: (注: 如果命令发送顺序发送不正确, 温度传感器不能工作, 还有在传输数据的过程中关键部分的延时也是要求很严格的, 在应用中如果命令顺序正确而温度值不能读出, 大多数情况下都是在数据传输过程中的时间延迟不正确, 但是本次试验中的这部分源代码已经给出, 要求同学们深刻理解温度传感器的使用方法, 做到在以后的应用中会用温度传感器测定环境温度。)

4. 在测量距离的过程中, 发射探头为什么不能一直都发送超声波, 而是采用瞬间发送然后停止发送的方法实现测量距离的功能。

答: 超声波传感器在测量较远距离的障碍物是非常有优势的, 然而在物体里测量系统非常紧的时候就显得有劣势了。这主要是因为超声波的测距过程是这样的, 首先在很短的时间内 (0.25ms 左右大约发射 10 个周期的超声波) 然后就关闭发射, 打开中断等待超声波的发射波。当反射波到来时经过发射电路后, 触发单片机的中断, 在发射和中断触发这段时间内就是超声波在物体与测距仪间的传播时间, 如果我们采用连续发射的方式, 在超声波发出并发射回来后, 超声波在接收探头处就是连续存在的, 这对于单片机来说此种形式的信息里是不存在距离信息的, 是不能完成测距的。

实验三 舵机控制实验

一、实验目的

- 1) 了解舵机的原理和控制方法。
- 2) 掌握单片机的定时器模块的使用。
- 3) 学会利用定时器来生成 PWM 波。

二、实验设备

PC 一台 (PC 上安装 MPLAB IDE 软件)、在线调试器 MPLAB ICD2、目标电路板、舵机一个。

三、实验原理

1. 舵机的工作原理

机器人的动力来源都是舵机, 嵌入式系统主要控制的对象就是舵机, 对舵机必须有一个

清楚的认识。舵机是一种位置伺服的驱动器，适用于那些需要角度不断变化并具有保持力矩的控制系统。舵机的工作原理是：控制电路板接受来自信号线的控制信号，控制电动机转动，电动机带动一系列齿轮组，减速后传动至输出舵盘。舵机的输出轴和位置反馈电位器是相连的，舵盘转动的同时，带动位置反馈电位器，电位器将输出一个电压信号到控制电路板，进行反馈，然后控制电路板根据所在位置决定电动机的转动方向和速度，从而达到目标停止。

控制信号由接收机的通道进入信号调制芯片，获得直流偏置电压。它内部有一个基准电路，产生周期为 20ms，宽度为 1.5ms 的基准信号，将获得的直流偏置电压与电位器的电压比较，获得电压差输出。最后，电压差的正负输出到电动机驱动芯片，决定电动机的正反转。当电动机转速一定时，通过级联减速齿轮带动电位器旋转，使得电压差为 0，电动机停止转动。舵机内部结构如图 7-41 所示。

电动机：为舵机提供动力。

减速箱：将电动机的高转速降低，并获得大的转矩。

电位器：为系统提供输出轴的反馈。

电子控制板：用来分析控制信号和反馈位置直接的关系，控制电动机正转、反转还是停机。真正的舵机如图 7-42 所示。

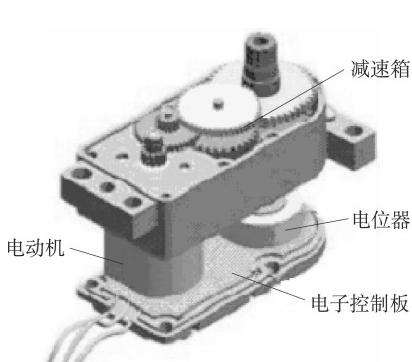


图 7-41 舵机内部结构

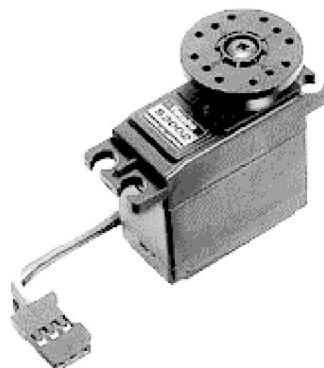


图 7-42 真实舵机图片

2. 舵机的 PWM 控制方式

采用的是日本 KONDO 公司生产的 KRS786ICS 舵机，这种舵机是专门为双足步行机器人设计的舵机，其具体参数见表 7-3。

表 7-3 KRS786ICS 舵机参数

| 舵机名称 | 尺寸/mm | 重量/g | 速度 | 力矩/N·cm | 周期/ms | 控制范围/(°) |
|-----------|------------|------|-----------|---------|-------|----------|
| KRS786ICS | 31×16×30.2 | 35 | 0.23s/60° | 77.42 | 8~20 | -90~90 |

舵机的控制信号的特征，如图 7-43 所示，舵机控制信号为周期从 8~20ms 可以任意选择的脉宽调制（PWM）信号，脉冲宽度为 0.7~2.3ms 的范围，对应于 -90°~90° 的转角范围，如图 7-44 所示。舵机控制信号的高电平为 3~6V，低电平为 0~0.8V。

与直流电动机相比较，舵机的控制要简单很多，主要是因为它有一个内置的控制器，于

是对嵌入式控制系统的要求就简单很多。只要嵌入式系统获取 PC 下载给它的数据之后，根据数据值向各个舵机发送周期性脉冲就可以了，只是脉宽不同而已。

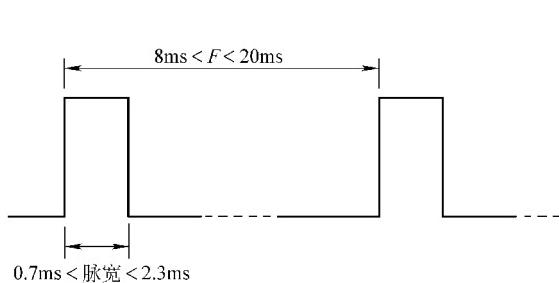


图 7-43 舵机的控制信号

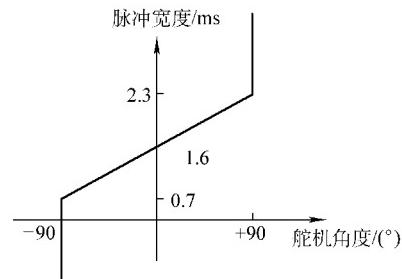


图 7-44 舵机的角度和脉冲宽度之间的关系图

今后为了运算和使用方便，把舵机的 -90° 规定为 0° ，把舵机的 90° 规定为 180° ，相当于把参考坐标系向左移动了 90° ，即如图 7-45 所示的本系统中使用的舵机角度和脉冲宽度之间的关系图，并且将舵机的控制信号定为周期是 20ms 的脉宽调制信号，其中脉冲宽度从 $0.5 \sim 2.5\text{ms}$ ，相对应舵盘的位置为 $0 \sim 180^\circ$ ，呈线性变化。也就是说，给它提供一定的脉宽，它的输出轴就会保持在一个相对应的角度上，无论外界转矩怎样改变，直到给它提供一个另外宽度的脉冲信号，它才会改变输出角度到新的对应的位置上。

3. 利用 PIC (PIC16F877A) 单片机定时器 1 生成 PWM 控制信号

PIC 单片机有 Timer0、Timer1、Timer2 三个定时器，双足步行机器人用 Timer0 做看门狗，以防止程序跑飞；用 Timer1 来定时比较，生成 24 路 PWM 波。下面就介绍一下 Timer1 (Timer0 和 Timer2 在一些用法方面和 Timer1 相似)。

Timer1 模块是由两个可读写的 8 位寄存器 (TMR1H 和 TMR1L) 组成的 16 位定时器/计数器。TMR1 寄存器 (TMR1H: TMR1L) 从 0000h 开始加 1 计数，一直加到 $FFFF\text{h}$ 后，再加 1 又循环回到 0000h 。如果允许 Timer1 中断，则定时器溢出的时候中断标志位 TMR1IF 置位，而产生 Timer1 中断。中断的使能/禁止可以通过置位/清除 TMR1IE 中断使能位来设置。

T1CON: Timer1 控制寄存器，用来控制 Timer1 的工作模式，其数据位定义如图 7-46 所示。

| | | | | | | | |
|-----|-----|---------|---------|---------|--------|--------|--------|
| U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| | | T1CKPS1 | T1CKPS0 | TIOSCEN | TISYNC | TMR1CS | TMR1ON |

bit 7

bit 0

图 7-46 Timer1 控制寄存器 T1CON 的数据位定义

bit 7:6 未定义，始终读为“0”。

bit 5:4

T1CKPS1: T1CKPS0: Timer1 输入时钟预分频选择

11 = 1: 8 预分频值；
 10 = 1: 4 预分频值；
 01 = 1: 2 预分频值；
 00 = 1: 1 预分频值。

bit 3

T1OSCEN：Timer1 内部振荡器使能位。

当 T1OSCEN = 1 时：打开内部振荡器，反相放大器工作，需外接晶体产生振荡时钟。

当 T1OSCEN = 0 时：振荡器被关闭。振荡器的反相器和反馈电阻被关。

bit 2

T1SYNC：Timer1 外部时钟输入同步控制位。

当 TMR1CS = 1 时：不同步外部时钟。

当 TMR1CS = 0 时：同步外部时钟。TMR1CS = 0 时 Timer1 使用内部时钟。

bit 1

TMR1CS：Timer1 时钟源选择位。

当 TMR1CS = 1 时：选择 T1OSO/T1CKI 引脚的外部时钟（上升沿计）。

当 TMR1CS = 0 时：选择内部时钟（FOSC/4）。

bit 0

TMR1ON：Timer1 使能位

当 TMR1ON = 1 时：使能 Timer1。

当 TMR1ON = 0 时：关闭 Timer1。

双足步行机器人用到的是 Timer1 的定时功能，将 TMR1CS (T1CON 控制寄存器 bit1) 清 0，选择 TMR1 工作在定时器模式。在这种模式下，定时器的输入时钟是内部时钟频率的 4 分频 ($f_{osc}/4$)。因为内部时钟总是同步的，所以同步控制位 T1SYNC (T1CON 控制寄存器 bit2) 此时不起作用。

Timer1 初始化：

```
void InitTimer ( void )
{
    OPTION |= 0x0F; //开启看门狗，分频为 1: 128 (Timer0)
    T1CON = 0;       //工作在定时模式
    TMR1IE = 0;     //关中断
    TMR1IF = 0;     //清除溢出标志
}
```

由于我们使用的舵机控制波形为高电平延时在 0.5 ~ 2.5ms 的方波，我们可以通过 Timer1 定时，比如我们要让某一引脚（如 RD1）输出高电平延时为 1.5ms，周期为 20ms 的方波，我们可以这样做：

- ①RD1 = 高电平；
- ②延时 1.5ms；
- ③RD1 = 低电平；

④延时 18.5ms；

⑤重复 I—IV。

所以我们只需要再写一个延时函数即可，代码如下：

```
void Delay ( U8 TH, U8 TL )
{
    TMR1H = TH;           //Load delay time High
    TMR1L = TL;           //Load delay time Low
    TMR1ON = 1;           //打开 timer1
    while ( ! TMR1IF )   //等待计数溢出
        CLRWDT ();       //清除看门狗
    TMR1IF = 0;           //清除标志位
    TMR1ON = 0;           //关闭 timer1
}
```

很显然，我们只要知道延时 1.5ms，单片机需要运行多少条指令就行了。下面就来详细计算一下：

PIC 单片机晶振为 20MHz，定时器的输入时钟是内部时钟频率的 4 分频 ($f_{osc}/4$)，也就是说，一条指令需要 $(4/f_{osc}) \text{ s} = 4/20\text{MHz} = 0.2\mu\text{s}$ ，定时 1.5ms 就需要执行 $1.5\text{ms}/0.2\mu\text{s} = 7500$ 条指令，所以定时器每次重装的计数初值为 $0xffff - 7500$ ，即 $\text{TMR1} = 0xffff - 7500$ ，从而得出：

$$\begin{aligned}\text{TMR1H} &= (0xffff - 7500)/256 = \text{HIGH}; \\ \text{TMR1L} &= (0xffff - 7500) \% 256 = \text{LOW};\end{aligned}$$

在程序中：

$$\begin{aligned}\text{TIME_CONST_uS_H}(x) &((0xffff - ((x) * (FOSC/4/1000000))) / 256) \\ \text{TIME_CONST_uS_L}(x) &((0xffff - ((x) * (FOSC/4/1000000))) \% 256)\end{aligned}$$

来表示 x 微秒的时间常数。

这样， $\text{Delay}(\text{TIME_CONST_uS_H}(\text{Angle}), \text{TIME_CONST_uS_L}(\text{Angle}))$ 就表示延时 Angle 微秒了。

我们要生成连续的周期为 20ms、高电平延在 0.5 ~ 2.5ms 的方波来控制舵机就很容易了（因为高电平延在 0.5 ~ 2.5ms，所以低电平延在 17.5 ~ 19.5ms），代码如下：

```
void PWM(U16 Angle)
{
    DUOJI_RUN;
    //高电平正时
    Delay(TIME_CONST_uS_H(Angle), TIME_CONST_uS_L(Angle));
    DUOJI_STOP;
    //低电平延时 17.5ms
    Delay(TIME_CONST_uS_H(10000), TIME_CONST_uS_L(10000));
    Delay(TIME_CONST_uS_H(7500), TIME_CONST_uS_L(7500));
```

```

//减去高电平时间,确保周期在20ms
    Delay(TIME_CONST_uS_H(2500 - Angle),
TIME_CONST_uS_L(2500 - Angle));
}

```

注意控制舵机必须是连续的方波，一个周期的方波是不行的，所以连续调用 PWM()就可以了。

四、实验步骤

1. 运行 MPLAB IDE

要启动 IDE，安装后双击桌面上的图标。

2. 创建项目

具体步骤如下：

- 1) 选择 Project→Project Wizard，单击下一步，选择器件 PIC16F877A，单击下一步。
- 2) 选择语言工具：在 Active Toolsuite 框中选 HI-TECH PICC Toolsuite，单击下一步。
- 3) 输入工程名和工程所在的路径，单击下一步；这里我们设工程名为 frist，存在路径 d:\ first\ 下。
- 4) 如果所建工程路径下已经有了写好的 C 文件，可以直接添加到工程中，若没有，直接单击下一步。
- 5) 单击完成，项目工程建立完成。

3. 编写程序

具体步骤如下：

- 1) 单击 File→New 新建一个文件，编写程序代码，完整代码如下（注意 RD1 引脚要设为输出）：

```

/*
; ****
; *          北京科技大学王志良类人机器人实验室
; *
; * _____ File Information _____ *
; *
; * File name : PWM.C
; * Function  : 熟悉 Timer 定时器的使用,熟悉使用定时器生成 PWM 波来控制
; *             : 舵机
; * Notice    : MCU:PIC16F877A
; *
; * _____ Latest version information _____ *
; * Author        : xuwenxue
; * Modified Date : 2006-8-21
; * Version       : v1.0.1

```

```

; * ----- History version infomation ----- *
; * Author : xuwenxue
; * Created date : 2006-8-21
; * Version : v1.0.0
; * -----
; *****

/
#include <pic.h> //包含 PIC 头文件

typedef unsigned char U8; //无符号 8 位(0 ~ 255)
typedef signed char I8; //有符号 8 位( - 128 ~ 127)
typedef unsigned int U16; //无符号 16 位(0 ~ 65535)
typedef signed int I16; //有符号 16 位( - 32768 ~ 32767)

#define FOSC 20000000 //定义晶振为 20MHz
#define INS 5 //定义 1μs 所需的指令数
#define DUOJI RD1 //舵机控制口定为 CH4 通道, 对应引脚为 RD1
#define DUOJI_RUN DUOJI = 1 //PWM 波的高电平, 高电平的时间对应舵机的转
                           //角, 0.5 ~ 2.5ms 对应 0 ~ 180°
#define DUOJI_STOP DUOJI = 0 //PWM 波的低电平

//#define TIME_CONST_uS_H(x)
((0xffff - ((x) * (FOSC/4/1000000))) / 256)
//#define TIME_CONST_uS_L(x)
((0xffff - ((x) * (FOSC/4/1000000))) % 256)

#define TIME_CONST_uS_H(x) ((0xffff - ((x) * INS)) / 256)
#define TIME_CONST_uS_L(x) ((0xffff - ((x) * INS)) % 256)

*****  

* Function: void InitTimer( void )
* Use: Initialize timers
* Input parameter: none
* Output parameter: none
***** /  

void InitTimer( void )
{
    OPTION1 = 0x0F; //开启看门狗, 分频为 1:128(Timer0)
    T1CON = 0; //工作在定时模式
}

```

```

TMR1IE = 0;           //关中断
TMR1IF = 0;           //清除溢出标志
}

/*****************/
* Function:       void Delay( U8 TH, U8 TL)
* Use:            Accurate delay function( using timer1 )
* Input parameter: TH: Value for TMR1H; TL: Value for TMR1L
* Output parameter: none
/*****************/
void Delay( U8 TH, U8 TL)
{
    TMR1H = TH;           //Load delay time High
    TMR1L = TL;           //Load delay time Low
    TMR1ON = 1;           //打开 timer1
    while( ! TMR1IF );   //等待计数溢出
    CLRWDT();             //清除看门狗
    TMR1IF = 0;           //清除标志位
    TMR1ON = 0;           //关闭 timer1
}
/*
void Delay( U8 TH, U8 TL)
{
#asm
    BCF STATUS,RP0
    movlw TH
    movwf TMR1H//Load delay time High
    movlw TL
    movwf TMR1L//Load delay time Low
    bsf T1CON,TMR1ON
loop btfsc PIR1,TMR1IF
    //asm(" clrwdt ")
    bcf PIR1,TMR1IF
    bcf T1CON,TMR1ON
#endifasm
}
*/
/*****************/
* Function:       void PWM( U16 Angle)
* Use:            PWM function( using timer1 )

```

- * Input parameter: Angle: Value for the position of the motor
- * Output parameter: none

```
***** /  

void PWM( U16 Angle)  

{  

    DUOJI _ RUN;  

    //高电平时间,对应舵机角度  

    Delay( TIME _ CONST _ uS _ H( Angle ) ,TIME _ CONST _ uS _ L( Angle ) );  

    DUOJI _ STOP;  

    Delay( TIME _ CONST _ uS _ H(10000) , TIME _ CONST _ uS _ L(10000) );  

    Delay( TIME _ CONST _ uS _ H(7500) , TIME _ CONST _ uS _ L(7500) );  

    //确保周期为 20ms  

    Delay( TIME _ CONST _ uS _ H(2500 - Angle ) ,  

TIME _ CONST _ uS _ L(2500 - Angle ) );  

}  

void main( void)  

{  

    U16 i = 0;  

    InitTimer();           //初始化定时器  

    TRISD &= ~ (1 << 1); //设置 RD1 为输出,以便输出 PWM  

    while(1)  

    {  

        i++;  

        if( (i < 250) && (i > 0) )  

            PWM(500); //高电平延时为 0.5ms,周期为 20ms 方波,舵机对应位置应为 0°  

        else if( (i < 500) && (i > = 250) )  

            PWM(1500); //高电平延时为 1.5ms,周期为 20ms 方波,舵机对应位置应为 90°  

        else  

            PWM(2400); //高电平延时为 2.4ms,周期为 20ms 方波,舵机对应位置应  

为 180°  

        if( i > 750 ) i = 0;  

    }  

}
```

我们通过改变 PWM 函数的参数在 500 ~ 2500 之间变动就可以实现舵机转角变化了, 可以在程序中设置一个 i, 每 20ms 加一, 到一定时间程序自己改变函数的参数就行了。实现舵机转速的变化也是这个道理, 只是改变的时间长短的不同。

- 2) 保存这个文件到 d:\first\ 文件夹下, 文件名为 first.c。
- 3) 添加 first.c 到工程中去, 右键单击 source files, 选择 “add files”, 找到 fisrt.c, 加进

去即可。

4) 保存工程，单击 File→Save Workspace。

4. 连接 MPLAB ICD2

具体步骤如实验一所示。

5. 调试程序

设置芯片的配置字、编译程序和下载程序等如实验一所示，下载完程序之后，将 ICD2 和目标演示板断开，将舵机连到 RD1 插口，我们就会看到舵机的舵盘转动起来，先转 90°，再转 90°，然后再回转 180°，用示波器看一下输出的波形对不对。

五、思考题

1. 舵机是如何控制的，和普通直流电动机有什么不同？

答：通过周期在 20ms 左右，高电平在一定范围的 PWM 波来控制，不同舵机高电平范围不同，具体看舵机参数。直流电动机也可以通过 PWM 来控制，所不同的是，直流电动机 PWM 是用来调速的，而舵机则是调位置的，事实上舵机内部也是个直流，不过在舵机内部还有个用来反馈电路板，通过 PWM 来调整所要转的角度。

2. 简述利用 Timer1 生成 PWM 的过程？

答：TIMER1 就是用来定时，先将某引脚置高（低），定时一段时间后再将该引脚置低（高），再定时一段时间将该引脚置高（低），如此反复即可。

3. Timer1 最多能定时多长时间？

答：这需要根据外围晶振的频率来定，我们知道单片机一条指令耗时： $(4/f_{osc})\text{ s}$ 。Timer1 是一个 16 位定时器，所以能从 0 加到 65535，也就是说最多能加 65536 个指令，当从 0 加到 65535 时也就定时了 $65536 * (4/f_{osc})\text{ s}$ ，这也是最大值。比如我们的晶振是 20MHz 的，所以最多能延时 13.1072ms，所以要注意程序中 (TIME_CONST_uS_H(x), TIME_CONST_uS_L(x)) 的 x 是有限制的， $X \leq 13107$ 。

实验四 机械结构设计与认识实验

一、实验目的

- 1) 认识双足步行机器人的自由度。
- 2) 掌握双足步行机器人机械结构。
- 3) 熟悉舵机初始角度和初始姿态对安装机器人影响。
- 4) 学会组装双足步行机器人。

二、实验设备

机器人组件一套、舵机 17 台、机器人控制器一个、若干螺钉。

三、实验原理与说明

1. 关于机器人自由度的说明

自由度从广义上来说就是在某一方向上能够旋转或移动。它是机器人性能中一个重要的

参数，和舵机联系在一起，一般来说，有多少个舵机就有多少个自由度。在设计机器人的机械结构时，要考虑到机器人的仿人特点。在每个关节处都要设立自由度，那样才能完成类似人的动作。比如设计机器人腿，至少要有 5 个自由度，分别是髋关节上的两个自由度、膝关节上的一个自由度、还有踝关节上的两个自由度，这样就可以完成类似人的前进、后退、抬腿、俯身等动作，如图 7-47 所示。

2. 机器人机械结构配件

图 7-48 所示的机器人，是将 17 台舵机以搭积木的方式搭成人形的。机体大部分是由舵机组件组成的，各个舵机之间是由一些钣金件连接而成。为了保证舵机之间能够有效地连接，同时 17 个舵机要组装成一个人形，需要有合理的机械配件。图 7-49 所示的就是机器人所需的配件。

</

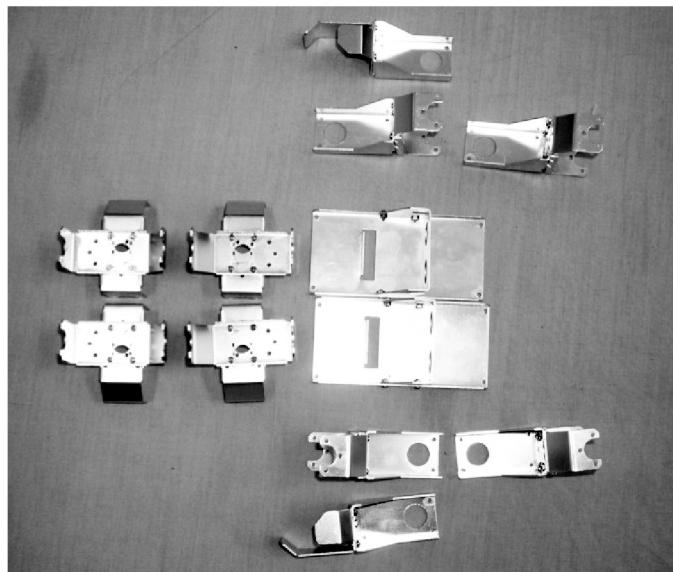


图 7-50 组装方式示意图

2) 用 M2 × 33mm 螺钉将配件和舵机组装在一起，如图 7-51 所示。注意在安装脚部舵机时，首先将舵机的外壳放到里面，然后将舵机的数据线从配件的侧面方孔穿过。这样安装才能不损坏数据线和舵机外壳。

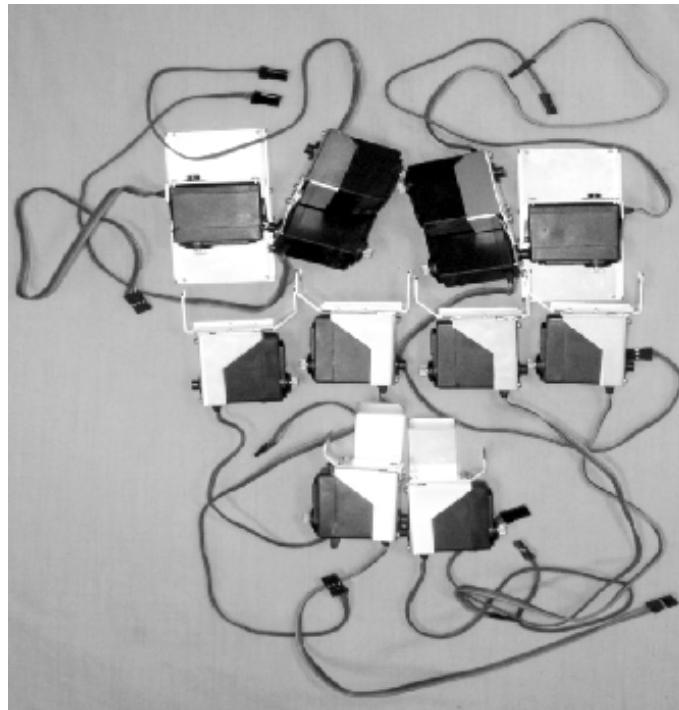


图 7-51 配件舵机组装示意

3) 组装机器人的两个上肢，每个上肢由两个舵机组成，具有2个自由度。安装前将舵机初始的角度设定在90°，这样有利于舵机有摆动的余地。因为此舵机最大动作角度是180°，当把安装角度设在90°时，与配件相配合不会发生运动干涉，可以顺利地完成一些指定的动作，如图7-52所示。

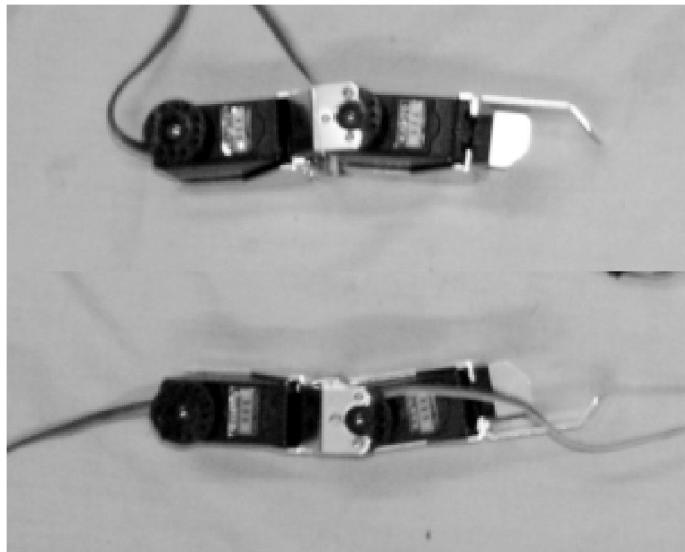


图7-52 机器人上肢

4) 组装机器人的两个下肢，每个下肢由4个舵机组成，具有4个自由度。安装前还是将舵机初始的角度设定在90°，另外注意在安装时，大腿的内侧有配件固定如图7-53所示。

5) 组装机器人躯干，由4个舵机组成，具有4个自由度，控制胳膊前后旋转和大腿左右摆动。安装前还是将舵机初始的角度设定在90°，许多配件是由螺钉固定，为了美观和拆卸方便，螺母均放在里面。在安装时，先从一面开始安装，这样有利于紧固螺钉。此处由于受到空间的限制，操作比较困难，那么在安装时一定要注意每个螺钉必须紧固牢靠，防止松懈，如图7-54所示。

6) 安装机器人控制器和机器头，此处安装比较容易。先将上肢两个舵机和机器头舵机的数据线放在控制器板的下面，这样有利于整体布置数据线。然后用3个螺钉将控制器固定在板件上。

7) 组装成一个整体，分别将上

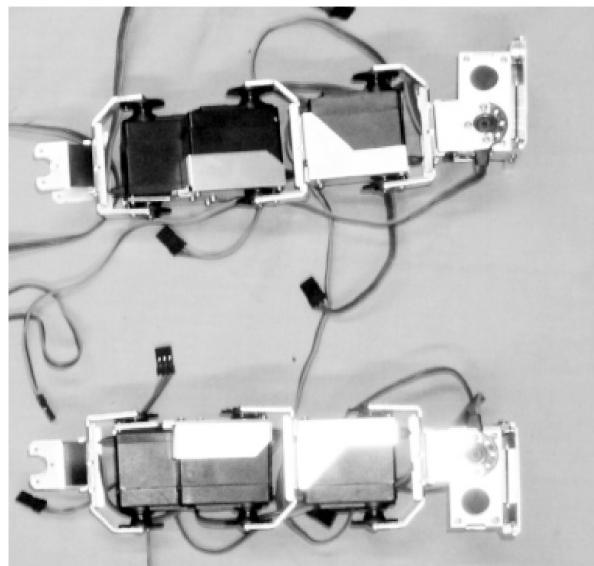


图7-53 机器人下肢

肢和下肢与躯干连接在一起。注意在安装时，大腿是处于并拢状态，胳膊处于伸直状态，如图 7-55 所示。

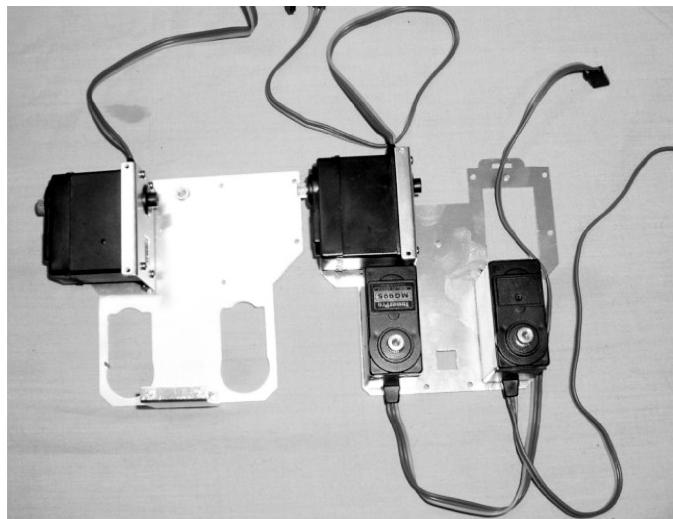


图 7-54 机器人躯干部

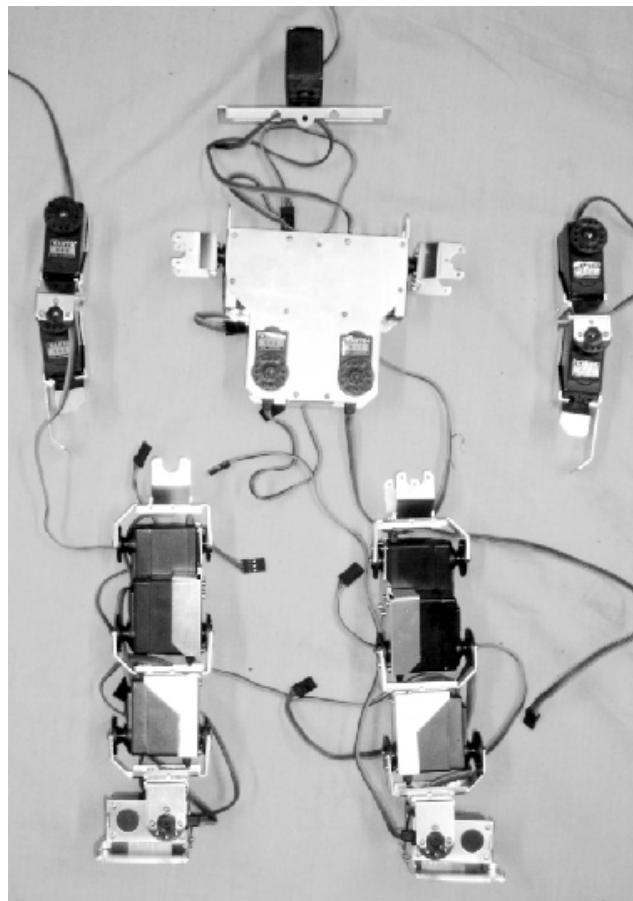


图 7-55 整体组装图

按照以上步骤就会完成机器人机械机构的组装。在组装前，首先要有整体的布局，然后再分步组装。相信通过以上的图片和文字，你一定会成功组装成一个具有 17 个自由度的双足步行机器人。

实验（二）舵机初始位置的角度和姿态对调试动作的影响

1) 为了表现机器人拟人特性，当通电以后要让机器人处于立正状态也称作初始位置 (Homeposition)。初始位置是机器人类似于人类，在重力的作用下，正处于能量最低的消耗状态。在每个机器人组装完毕后，首先调试出的就是初始位置状态。图 7-56 和图 7-57 分别是机器人处于初始位置的正面和侧面图。

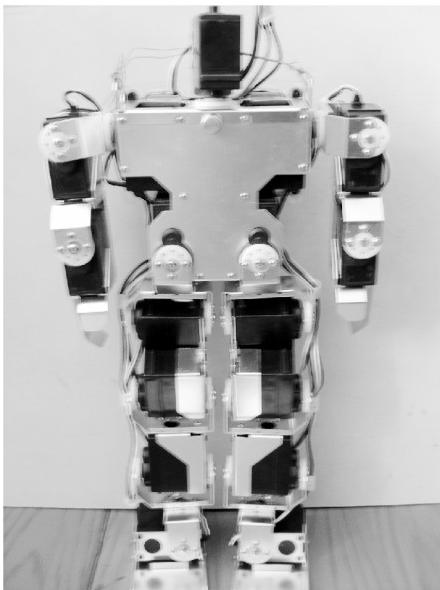


图 7-56 机器人正面图

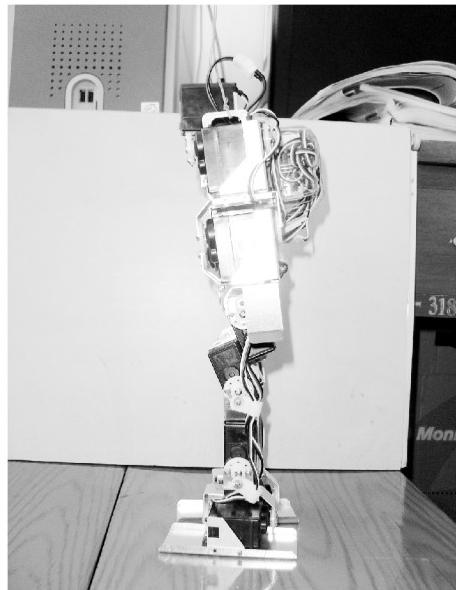


图 7-57 机器人侧面图

2) 舵机的初始角度是在安装时各个舵机所处的角度。一般的情况下，我们都将舵机调节为 90° ，这样便于安装。安装时，大家可能有这样的体会：不让胳膊处于伸直状态，让它处于垂下状态和躯干上的舵机连接。如果这样就会发现，刚安装的胳膊不能够举过头部，需要重新安装。因为刚才安装的两个舵机发生了运动干涉，各个舵机具有耦合关系，虽然两个舵机均处于 90° 的状态，但是安装以后两个舵机的活动空间发生了变化。读者可以按照舵机的不同初始角度值来组装机器人，看看产生何种结果？

3) 机器人的初始姿态是指将各个舵机连接在一起时机器人所处的姿态。在将胳膊和大腿与躯干连接时，舵机的初始角度都调到 90° ，但是没有按照图 7-55 的姿态安装，可能就会产生某一个舵机活动干涉，不能够完成类人动作。读者可以按照不同的姿态安装机器人看看产生何种结果。

五、思考题

1. 说一说本实验 17 个自由度机器人的自由度分别用于控制哪个方向的？

答：

(1) 上肢有 6 个自由度，分别是肩部两个自由度和肘部一个自由度。肩部两个自由度分别控制胳膊前后旋转和上下摆动，肘部一个自由度控制上下摆动。左右肢对称共 6 个自由度。

(2) 下肢共有 10 个自由度，分别是脚部两个自由度、膝关节一个自由度、髋关节两个自由度。脚部两个自由度分别控制脚前后和左右的摆动，膝关节一个自由度控制小腿前后摆动，髋关节两个自由度分别控制大腿前后和左右摆动。左右肢对称，共 10 个自由度。

(3) 头部一个自由度控制头部的左右旋转。

2. 说出 Homeposition 值和初始角度值的区别？

答：Homeposition 值是指当机器人通电以后，要让机器人处于立正状态时，17 个舵机的角度值。它是在每个机器人组装完毕后，首先调试出的，然后才能将机器人用于实践操作。

初始角度值是在组装机器人时，为了使 17 个舵机运动不发生干涉而且具有良好操纵性，将 17 个舵机的角度值分别调到 90°，就是初始角度值。它是在机器人组装时，各个舵机和连接件之间的相对角度。

3. 请说出舵机的初始角度为 90° 和机器人的初始姿态如图 7-55 所示是不是安装机器人唯一方法？为什么？

答：舵机的初始角度为 90° 和机器人的初始姿态如图 7-55 不是安装机器人唯一方法。因为舵机和连接件之间的相对角度可以是任意值，不一定是 90°，但是为了容易看到两者之间的角度，也就是垂直或平行关系，一般将舵机在安装时调到 90°。

实验五 简单动作调试实验

一、实验目的

- 1) 掌握双足步行机器人上位机控制软件的操作。
- 2) 熟悉双足步行机器人上位机软件如何控制舵机的转动。
- 3) 学会机器人初始位置 Homeposition 的调试。
- 4) 学习使用上位机控制软件，完成简单动作——招手的调试，体会速度控制方式不同对动作的影响。

二、实验设备

PC 一台（装有 VC 程序，双足步行机器人上位机控制软件）、无线收发装置（包含无线收发盒、串口通信线和 USB 数据线）一套、类人双足步行机器人一台，7.4V 充电电池一块、电池充电设备（包含 12V/1A 稳压直流电源和平衡充电器）一套。

三、实验原理与说明

关于 Homeposition 的说明：

Homeposition，是机器人良好站立、腿部各舵机受力均匀时的姿态名称。每一个动作数据的开始和最后都是使用的这个姿态。机器人在安装完毕之后，应首先调试出 Homeposition。图 7-58 和图 7-59 分别是机器人处于 Homeposition 的正面和侧面图。

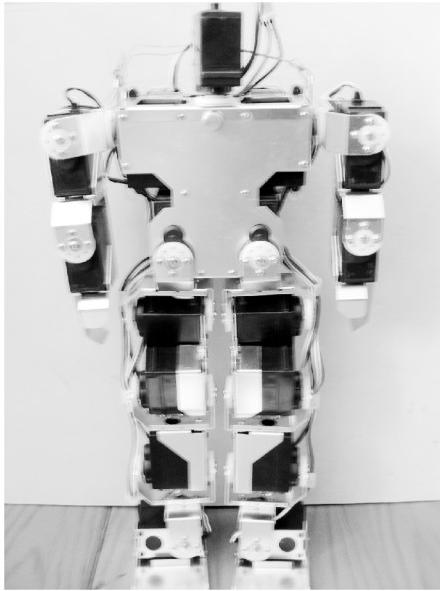


图 7-58 Homeposition 正面图



图 7-59 Homeposition 侧面图

因为每一个动作数据的开始和最后都是使用 Homeposition 这个姿态，它的调试至关重要，调试过程要遵循以下原则：

机器人良好直立、腿部各舵机受力均匀，如图 7-59 所示肩关节、髋关节、膝关节、踝关节四个舵机侧面的舵盘在同一个直线上，两条腿调试结束后各对应舵机在同一平面上。实验步骤的实验（一）中具体介绍如何调试 Homeposition。

四、实验前准备

- 1) 检查机器人电池电量是否充足，用万用表测量电池放电接头两端电压，若小于 6V，则需要更换电池，并对原有电池充电。

- 2) 检查机器人各舵机连线，确保完好无损，若发现有连线断接，或者刮破外绝缘皮等现象，则要先进行简单处理，如重新接线、贴绝缘胶布等措施。

- 3) 检查舵机连接件是否变形，若有变形，则需更换该部件。

- 4) 机器人检查完好之后，确保一切正常才可开电，否则后果自负。

- 5) 如图 7-60 和图 7-61 所示连接无线收发装置，它共有两条连接线，一端按照图示方式连接，另外一端分别接计算机的串口 1 和 USB 接口，连接好之后则指示灯亮，打开上位机软件，界面如图 7-62 所示，如图 7-63 所示单击菜单栏“设定串口参数”，



图 7-60 无线发射装置

弹出如图 7-64 所示界面，按照图示进行设置，单击“OK”，退出软件并重新启动。启动软件后，用鼠标拖动动作数据滑块中的任意一个，则该装置的指示灯将会如图 7-65 所示不停闪烁，说明反应连接无误，运行良好。否则，检查连线并重新连接，直到正常为止。



图 7-61 无线发射装置连接方法

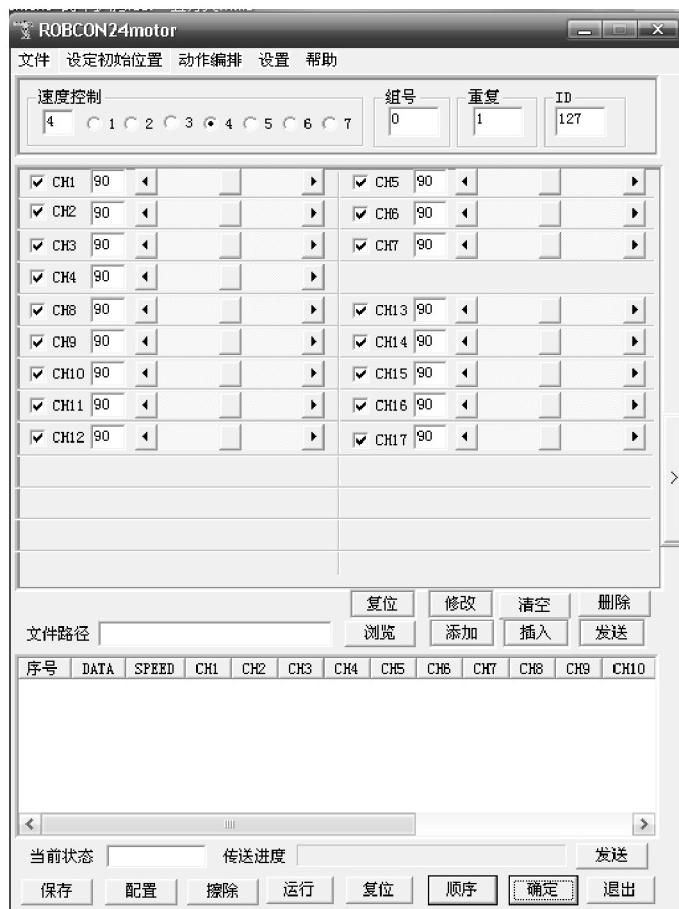


图 7-62 上位机软件界面



图 7-63 菜单栏示意图



图 7-64 串口设置对话框



图 7-65 无线发射装置正常运行

五、实验步骤

实验（一） 调试并存储 Homeposition

1) 如图 7-66 所示将机器人背部面板上两个小开关同时拨到图示位置，使机器人处于无

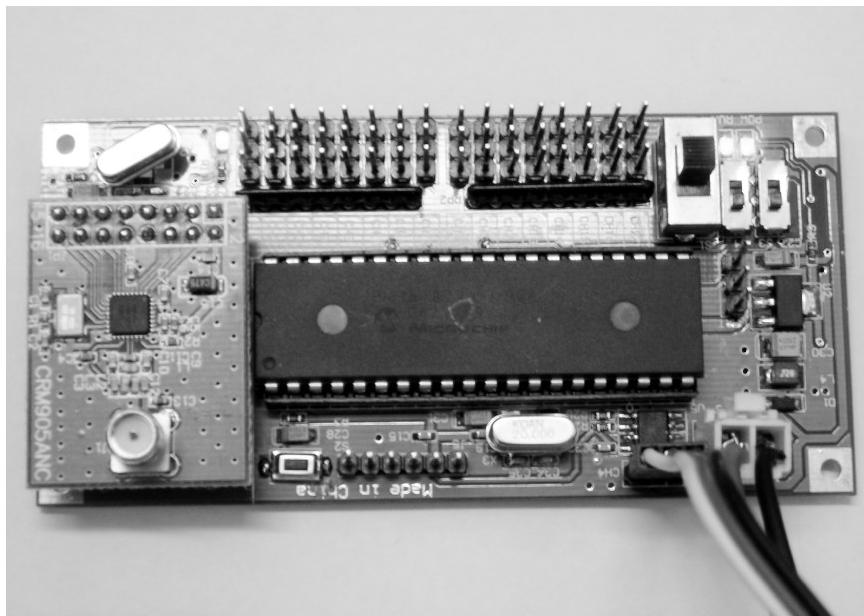


图 7-66 机器人控制电路板

线控制（或同时拨到相反位置并将主机的串口线接到电路板上采取有线方式调节），在此采用无线方式，打开机器人电源，控制板上三个指示灯均亮。

2) 清楚掌握各舵机转动角度增大和减小时机器人的动作变化，来填写表 7-4，只有掌握这个表，才能自如地通过上位机控制软件来调节机器人的动作。注意我们都以从背面来观察机器人的角度来得到下面各个舵机在度数变化时的转动方向。填写之后通过上位机程序上的滚动条来验证结果。

注意：在表 7-4 中的第 1 个与第 5 个舵机的栏中填入方向为“前后”，第 2、3、6、7 个舵机的栏中填入方向为“内外”，第 4 个与第 8、13 个舵机的栏中填入方向为“左右”，第 9、10、11、14、15、16 个舵机的栏中填入的方向为“前后”，第 12 个与第 17 个舵机的栏中填入“顺逆时针转动”。请同学们认真填写，之后通过上位机程序上的滚动条来验证结果。

表 7-4 各舵机度数变化的转动方向（从背面看各舵机变化情况）

| 舵机号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|-----|------|------|------|------|------|------|------|-------|-------|-------|-------|---------|-------|-------|-------|-------|---------|
| 增大 | 大臂后摆 | 大臂向里 | 小臂向里 | 头部左摆 | 大臂前摆 | 大臂向外 | 小臂向外 | 髋关节前摆 | 膝关节左摆 | 膝关节后摆 | 踝关节后摆 | 踝关节顺时针转 | 髋关节左摆 | 髋关节后摆 | 膝关节前摆 | 踝关节前摆 | 踝关节顺时针转 |
| 减小 | 大臂前摆 | 大臂向外 | 小臂向外 | 头部右摆 | 大臂后摆 | 大臂向里 | 小臂向里 | 髋关节右摆 | 膝关节后摆 | 膝关节前摆 | 踝关节前摆 | 踝关节逆时针转 | 髋关节右摆 | 髋关节前摆 | 膝关节后摆 | 踝关节后摆 | 踝关节逆时针转 |

3) 根据图 7-67 的机器人舵机对应图，通过上位机控制软件的滑块来调节舵机的度数（图 7-67 中机器人各舵机编号与图 7-68 动作数据滑块一一对应，例如 1 号对应与 CH1，依次类比下去），调试出机器人初始位置 Homeposition。

调试机器人初始位置的方法：

①机器人电路板初次打开电源，各舵机处于保持状态，因此打开电源之前可先手动调整各舵机位置让其有较大的活动范围，避免上电之后调整初始位置时发生堵转。

②读取参考初始位置文件（后缀名为 .hos），并设置为现有机器人的初始位置，此时机器人有了粗略的初始位置，由于机器人机械结构的差异，需要对其进行微调。

③调整髋关节左右方向移动的舵机（8 号和 13 号），使得两条腿平行且为竖直方向，同时注意两条腿中间有 5 mm 左右的间隙，避免在做动作时出现摩擦。

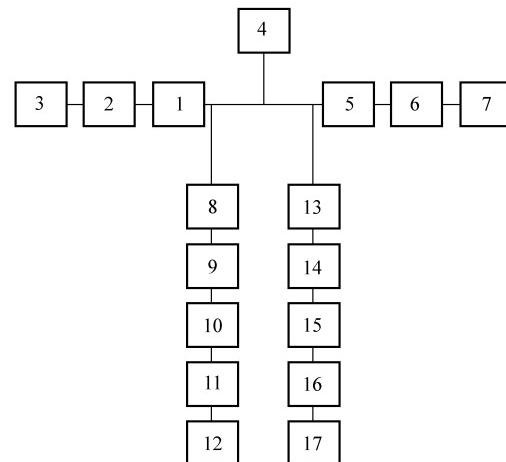


图 7-67 从机器人背面观察的舵机标号

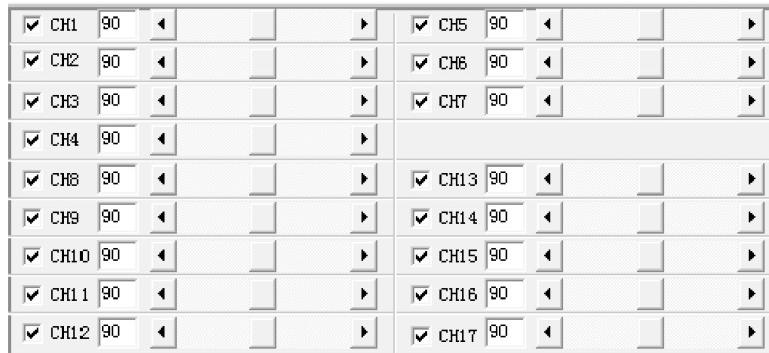


图 7-68 动作数据滑块

④调整踝关节前后转动的舵机（11号和16号），使其与脚底板垂直，接着调整膝关节舵机（10号和15号），使其与踝关节舵机两舵盘连线与脚底板垂直，依次向上调整膝关节和髋关节前后方向转动的舵机（9号和14号），最终使得以上3个舵机，再加上肩部舵机（1号和5号），总共4个舵机的舵盘在一条直线上，并且确保与脚底板垂直。

⑤调整肘关节舵机（3号和7号），使手臂伸直成一条直线，调节肩关节两对舵机（1号和2号、5号和6号）使手臂为竖直方向下垂，调整头部舵机，使其处于正视位置并能左右各摆动90°。

4) 最后，观察整个机器人成立正姿势，既不前俯也不后仰，平稳站立，手臂下垂，左右对称，则机器人初始位置调试成功。

5) 如图7-69所示，单击“设定初始位置—设定机器人的初始位置”，将机器人的初始位置设定成刚刚调试完成的Homeposition。



图 7-69 菜单栏中的设定初始位置项

6) 单击“设定初始位置—保存初始位置数据”，设置好文件名，后缀为“.Hos”。

实验（二）设计机器人“招手”动作

1) 单击“添加”按钮，将Homeposition数据放置在数据列表框中，作为将要调试的招手动作的开始。

2) 先调试手臂抬起动作，拖动动作数据滑块中的CH1（对应于机器人的1号舵机），使机器人手臂上抬，与竖直方向成30°夹角，达到如图7-70所示的招手的动作一，然后单击“添加”按钮，这时数据将自动添加到数据列表框中，位置位于第2行，第1行为Homeposition。

3) 再调试招手动作，拖动动作数据滑块中的CH3，使机器人前臂向内弯曲成120°，达

到如图 7-71 所示为招手的动作二，然后再单击“添加”按钮，这时这一行的数据应该再添加到第 3 行，于是已经完成了 3 行动作数据的调试。

4) 由于招手这个动作是周期重复的，可以将招手动作一与招手动作二作为一个组合，进行周期性的重复，例如双击数据列表框的第 2 行，然后单击“添加”则生成第 4 行动作，双击第 3 行，然后添加为第 5 行，依次周期性的重复第 2 行和第 3 行动作。

5) 重复多次之后，即可做出最后一行的数据，调试机器人的动作有这样一条规则：“任何动作的第一行和最后一行都要是 Homeposition”。因此最后需要做的就是把 Homeposition 的数据添加到最后一行。双击数据列表框的第一组 Homeposition 数据，这样该行的数据都显示在动作数据滑块上，然后单击“添加”按钮，将 Homeposition 添加到列表框的最后一行，作为招手动作的最后一组动作。

6) 动作调试完毕后，就要给机器人发送数据了。如图 7-72 所示对操作界面中的“组号”进行配置，默认为 0，如果不是一套动作要更改组号，以免覆盖已有动作数据。接着如图 7-73 所示单击下端的“发送”按钮，将已调试好的招手动作发送至下位机，出现对话框后单击“确定”，之后单击“运行”按钮（同样在单击之前也要选择相应的组号），机器人将完成招手动作。

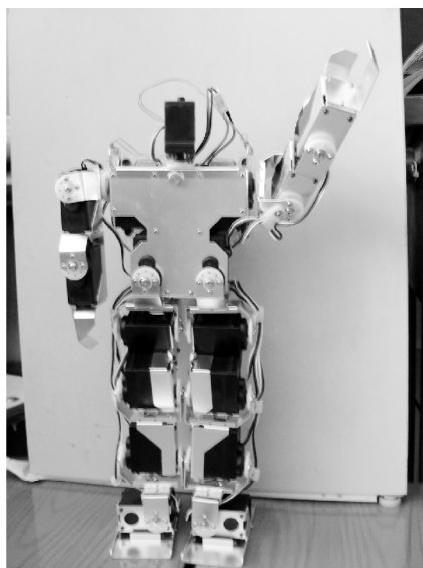


图 7-70 招手动作一

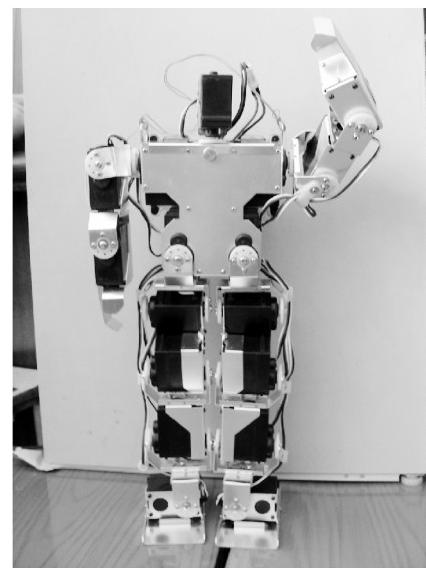


图 7-71 招手动作二



图 7-72 组号配置图示

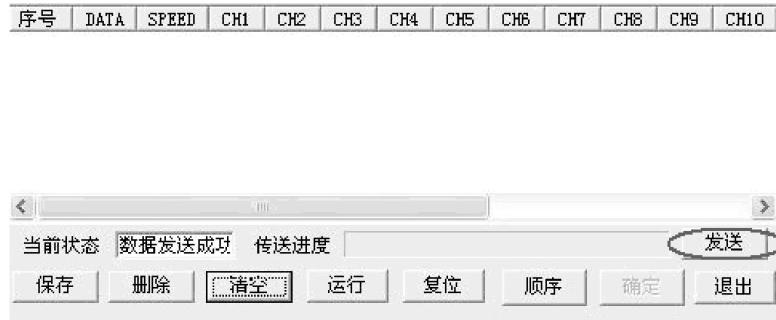


图 7-73 发送数据图示

7) 修改动作的速度,例如将速度改为6或者7(默认为4),按上述步骤重新调试一遍招手动作,观察不同速度对机器人动作的影响。

实验(三) 设计机器人“蹲下扩胸”动作

1) 单击“添加”按钮,将Homeposition数据放置在数据列表框中,作为将要调试的蹲下扩胸动作的开始。

2) 先调试机器人下蹲动作。手持机器人腰部,拖动动作数据滑块中的CH11和CH16(对应机器人踝关节11号和16号两舵机),使两脚平面平行,且使双腿与两脚平面成45°夹角;再拖动CH15和CH10,使膝关节15号和10号两舵机与脚平面垂直;进一步拖动CH14和CH9,使本来向后倾斜的上半身挺直,也就是与脚平面垂直。到此整个蹲下动作完成,单击“添加”,成为第2行数据。在此应注意重心问题,若太前倾则机器人会倒下;若后仰则机器人就不美观。

3) 接着调试机器人扩胸动作,该动作只用改变了胳膊的舵机转动角度,而下半身仍保持原有姿态。拖动CH1和CH5,使双臂向前平举,接着拖动CH3和CH7,使前臂向内弯曲,两个肘关节成120°,到此扩胸动作第1步完成,添加为第3行动作。

4) 第4行的动作是扩胸的第2步,拖动CH2和CH6,使上臂与肩成120°夹角,添加为第4行动作。

5) 第5行的动作是扩胸的第3步,即拖动CH2、CH3、CH6、CH7,将双臂打开成水平一线,添加为第5行动作,而腿部舵机转动角度没有改变。到此整个扩胸动作完成。

6) 从第6行开始,各行都作是对第3、4、5行扩胸动作的周期重复。例如,双击数据列表框中的第3行,单击“添加”,生成第6行动作,然后双击第4行同理添加为第7行,双击第五行添加为第8行。类似多次重复,生成第9、10、11行等。

7) 当做完扩胸运动之后,机器人放下双臂,腿仍然保持蹲下的姿势,这与第2行的下蹲动作相同,双击数据列表框的第2行,单击“添加”按钮,生成新的一行动作。

8) 最后一行仍然是Homeposition,双击第1行,单击“添加”按钮,生成最后一行,这样所有的动作都完成了。

9) 为了检验动作的好坏,需要先进行单步检验,首先机器人上电,连接好串口,将刚才得到的蹲下扩胸的数据,从头到尾,双击每一行,每当双击该行时,机器人就会实时地做出相应的动作。

10) 当单步检验都通过时，我们再将整个动作完整地发送到下位机。方法与实验二招手动作中图 7-74 相同。

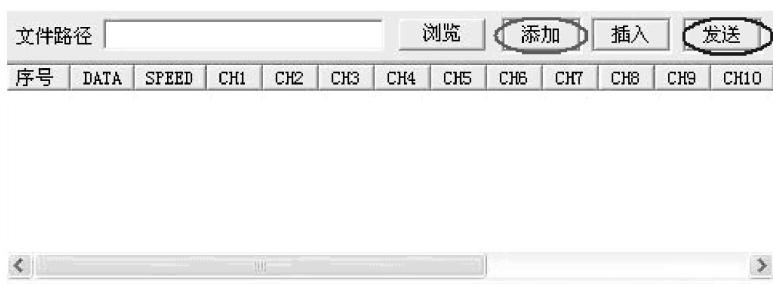


图 7-74

六、思考题

1. 请列出用上位机控制软件的滚动条进行简单动作调试的优缺点。

答：当用上位机控制软件的滚动条进行简单动作调试时，

优点：调试方法简单易行，不需要具有复杂的技术和编程知识，也可对机器人进行动作操作，适用于各个年龄段的用户。在机器人已有动作的前提下，可采用临摹法，一步一步调出，使得新的机器人也具有同样的动作，简单方便。

缺点：效率不高，所有的动作数据要逐行写入。精确度不够，因为这种方法大部分是通过观察体会，难免会受到主观因素和环境因素的限制，导致动作不够规范。特别是如：“行走”“上楼梯”等精确度要求很高的动作，用该方法实现难度较大。

2. 我们有 1~7 共七个速度，哪个速度舵机速度是最快的？具体调试动作时，根据什么选取合适的速度位？

答：我们有 1~7 共七个速度，7 号速度是速度最快的。反之，1 号速度是速度最慢的。那么，4 号速度是我们默认的普通速度。

具体调试动作时，要根据具体动作的情况来决定速度的大小。如果编写动作是普通动作，对稳定性要求不高时可以选择 4 号普通速度，或 5 号及 6 号的高速度，这样机器人的动作速度加快，增加了机器人动作的美感。对稳定性要求很好时，可以选择 2 号或 1 号低速度，来保证动作的顺利进行。总而言之，视具体动作的具体要求，选择合适的速度。

3. 上位机控制软件中“添加”和“插入”按钮的作用有何区别？

答：如图 7-74 所示，添加按钮的作用是将滚动条当前显示的数据，添加到列表框的最后一行，若再添加，就会添加到刚才的那行的下一行，简而言之，添加按钮永远将数据加到最后一条。插入按钮的作用是将滚动条当前显示的数据，插入到列表框光标所停的那一行的前一行，数据列表框光标所停的那一行可以通过单击鼠标左键来选择，注意不要用鼠标双击，如果双击的话，动作数据滑块的就会同步更新为数据列表框中选中的哪一行的数据，如果正在保持良好通信时，则会执行该行动作。

4. 为什么在调动作时，要先保证单步运行顺利，再将数据发送到下位机，进行整体运

行。如果跳过单步运行，直接整体发送和运行，会有什么后果？

答：如果跳过单步运行，当然可以直接整体发送和运行。但是带来的结果都不是我们所预料的那样，由于对于动作本身，一个连续的动作都是由各个静态的动作组成，如果单步的某个动作不是很稳定，试问整体动作一起运行怎么会成功呢。

所以我们规定在调动作时，除一些极其简单动作之外，都要先保证单步运行顺利，再将数据发送到下位机，进行整体的数据运行。这样才能保证动作的稳定，但这样也不是充分的，还要看连贯的动作是否顺利进行。这里要指出的是：“一般来讲，单步运行不稳定，连续整体运行一定不稳定；但单步运行稳定，连续整体运行也不一定稳定。”因为这里涉及单步动作衔接的问题，就好比两个没有关联的动作，每个都通过了单步运行，但整体运行一定不成功，因为它们没有衔接。所以在单步运行之后，还要考虑动作衔接的问题，这样才能保证我们所调试出的动作在整体运行时达到万无一失的效果。

实验六 动作数据编程实验

一、实验目的

- 1) 进一步熟悉上位机软件对类人双足步行机器人的控制。
- 2) 掌握机器人的动作数据得来的方法以及镜像法的思想。
- 3) 巩固学生 C++ 编程的能力。

二、实验设备

PC 一台（装有 VC 程序，双足步行机器人上位机控制软件，机器人动作数据文件），无线通信模块/串口通信线，类人双足步行机器人一台，电源（7.4V 锂电池），电池充电设备一套。

三、实验原理与说明

当我们在编排机器人动作的时候，经常会发现一些动作是和前面所做的动作有对称的关系，比如：当我们编排机器人上楼梯的动作，如果在我们已经成功地完成机器人用右脚踏上楼梯的第一个台阶的基础上，我们就可以用镜像法来精巧地得到对称的动作数据。由于下一个动作肯定是要让机器人在右脚的支撑下，伸出左脚踏上第二个台阶，这样的动作就是完全地与前面一个动作相对称，我们就可以利用镜像法，计算并调试出动作数据。

下面介绍一下镜像法：

对于两个动作，如果左右完全对称（第一个动作左边的形态与第二个动作右边的形态完全对称，第二个动作就好像是第一个动作在镜子中的镜像，故称此法为镜像法），则只需先调通一个动作即可。对于另一个动作，利用镜像法就可调出。这里需要注意的是前后动作是否对称，都是以 Homeposition 的动作作为参照。

当了解到镜像法的原理之后，我们所要做的就是找出左右舵机的对应关系，这在我们上次的舵机调试的实验课上已经得到了，这里我们就可以直接运用。

当然，即使我们了解了舵机的镜像对应关系，也知道如何改前面的动作数据成为新的动

作数据，但如果在动作特别复杂，数据特别长的条件下，我们的工作量难免就要大很多，为了提高效率，我们可以考虑编写一个简单的 C++ 程序，来实现对新的动作数据的计算。

最后将我们所得到的镜像的动作数据存放起来，并用双足步行机器人上位机控制软件来发送这些数据，机器人顺利完成动作，方可达到要求。

四、实验步骤

本实验是利用 VC 软件平台，编写镜像法程序，在给出机器人招左手的第一个动作的前提下，通过镜像法完成机器人招右手的第二个动作。

- 1) 首先在实验前要清楚哪两个号的舵机互相对称，这是要做镜像法的前提。
- 2) 清楚掌握各舵机转动角度增大和减小时机器人的动作变化，这里由于我们在上次的舵机调试的实验课里已经得到了结果，见表 7-5。同时注意都以背面观察机器人来得到相应各个舵机的转动方向（从背面看各舵机变化情况）。

表 7-5 各舵机镜像对应关系表

| | | | | | | | | | |
|----------|----|---|---|---|----|----|----|----|----|
| 对应 关系 | 舵机 | 1 | 2 | 3 | 8 | 9 | 10 | 11 | 12 |
| | 号码 | | | | | | | | |
| | 舵机 | 5 | 6 | 7 | 13 | 14 | 15 | 16 | 17 |
| | 号码 | | | | | | | | |

3) 清楚掌握舵机的镜像对应关系与舵机的转动方向之后，我们就可以掌握当机器人做左右对称动作时，各个对应的舵机所转的角度的关系，从而可以为写镜像法程序打好基础。通过表 7-4 与表 7-5 可以了解到，当机器人抬左手时，某号舵机转动了一个角度；然后机器人复位，之后再抬右手且姿势刚好与左手所抬的高度一样时，镜像对应的舵机转动了相同的角度，只是存在正负号的问题。请同学们将所有两两对应的舵机都通过上位机控制机器人，来得到镜像对应的舵机转动角度正负号的关系，然后填入表 7-6，因为如果弄错一个正负号的话，结果就大相径庭了（在表中填入舵机转动角度的正负号，其中，正号表示舵机角度增大，负号表示舵机角度减小）。

表 7-6 左右对称动作时各镜像舵机转角度正负号对应关系表

| | | | | | | | | | |
|--|----|---|---|---|----|----|----|----|----|
| 当左 右动 作对 称时 舵机 转动 度数 正负 关系 | 舵机 | 1 | 2 | 3 | 8 | 9 | 10 | 11 | 12 |
| | 号码 | | | | | | | | |
| | 舵机 | 5 | 6 | 7 | 13 | 14 | 15 | 16 | 17 |
| | 号码 | | | | | | | | |

4) 这样我们的准备工作大致就完成了。一套完整的镜像动作以 Homeposition 开始，此时舵机的转动角度为 H；最终达到的动作状态所对应的舵机转动角度为 A；中间过渡过程中

每一个动作状态所对应的舵机转动角度为 R 。请同学们将三者的关系填入机器人动作时的度数公式：

$$\underline{\quad} + \underline{\quad} = \underline{\quad} \quad (7-1)$$

5) 由公式 (7-1) 来推导公式 (7-2)。公式 (7-2) 已知机器人实际动作以后，最终达到的原像舵机转动角度 A_1 (8 个舵机的度数) 与开始动作时原像舵机的 Homeposition 转动角度 H_1 (8 个舵机的度数)，机器人做对称动作后的镜像对应舵机的转动角度 A_2 (另外 8 个舵机的度数) 与开始动作时镜像舵机的 Homeposition 转动角度 H_2 (另外 8 个舵机的度数) 四者的关系，请同学们将四者的关系填入机器人动作对称时前后度数之间的对应关系式：

$$\underline{\quad} - \underline{\quad} = \underline{\quad} - \underline{\quad} \quad (7-2)$$

之后我们就得到重要的式 (7-2) 的推论，机器人动作对称时舵机转动角度之间的镜像关系式：

$$A_2 = \underline{\quad} + \underline{\quad} - \underline{\quad} \quad (7-3)$$

6) 下面我们开始利用 VC 软件平台来编写镜像法函数。主要思想就是在已经给定的动作数据的条件下，利用公式 (7-3) 计算出对称的动作数据。这个计算通过编写函数来实现，避免每行数据用手算的工作量，达到提高效率的作用。我们在这里要求同学利用 C++ 函数，直接在 VC 里生成 C++ 源文件，而无须利用 MFC。如图 7-75 所示。

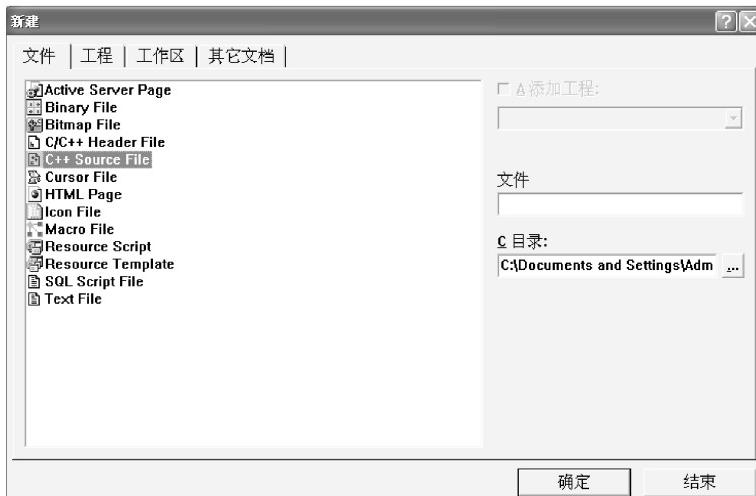


图 7-75 创建 C++ 源文件界面

7) 直接生成 C++ 文件，其中文件名随意起，之后在源文件中进行 C++ 编程，主要是利用函数实现动作数据的计算。这里不用子函数，只利用 main 函数实现其功能就可以。由于我们利用 C++ 的输入输出流，所以要加入 <iostream.h> 头文件。如图 7-76 所示。

在编写镜像法程序前，首先我们要填写两个表。打开上位机软件，在给定的机器人动作数据文件中调出该台机器人的初始位置。将机器人在初始位置时各舵机的转动角度值填入表 7-7 和表 7-8。

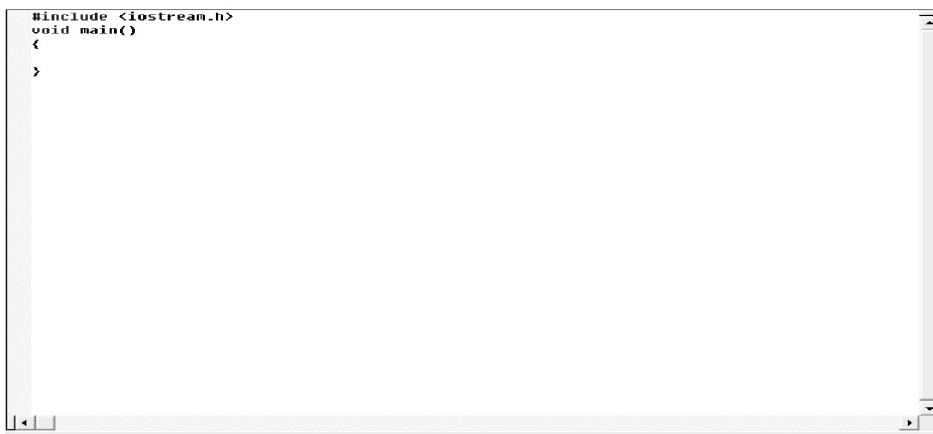


图 7-76 操作界面

表 7-7 机器人初始位置时各舵机的角度值

| 舵机号 转动角度 | 1 | 2 | 3 | 8 | 9 | 10 | 11 | 12 |
|-------------|---|---|---|---|---|----|----|----|
| H1 | | | | | | | | |

表 7-8 机器人初始位置时各舵机的角度值

| 舵机号 转动角度 | 5 | 6 | 7 | 13 | 14 | 15 | 16 | 17 |
|-------------|---|---|---|----|----|----|----|----|
| H2 | | | | | | | | |

//////////////////////////////镜像法函数////////////////////////////

```
#include <iostream.h>
void main( )
{
int a,b,c,d,e,f,g,h;
cout << "请输入各舵机起始度数:" << endl;
cout << "1/5:";           //舵机号码
cin >> a;                //用户输入该舵机的度数
cout << "2/6:" ;
cin >> b;
cout << "3/7:" ;
cin >> c;
cout << "8/13:" ;
cin >> d;
cout << "9/14:" ;
cin >> e;
cout << "10/15:" ;
```

```

cin >> f;
cout << "11/16:" ;
cin >> g;
cout << "12/17:" ;
cin >> h;
a = ( H1[ 1 ] + H2[ 5 ] - a );           //利用公式(5-5)A2 = H1 + H2 - A1 其中 H1
                                            [ 1 ] 和 H2[ 1 ] 分别是对应的两个 Homeposition
b = ( H1[ 2 ] + H2[ 6 ] - b );
c = ( H1[ 3 ] + H2[ 7 ] - c );
d = ( H1[ 8 ] + H2[ 13 ] - d );
e = ( H1[ 9 ] + H2[ 14 ] - e );
f = ( H1[ 10 ] + H2[ 15 ] - f );
g = ( H1[ 11 ] + H2[ 16 ] - g );
h = ( H1[ 12 ] + H2[ 17 ] - h );
int data[ 8 ] = { a,b,c,d,e,f,g,h } ;      //定义数组
int i = 8;
for( i = 0 ; i < 8 ; i ++ )                //保证算过的舵机度数在 1 ~ 179 之间
{
    if( data[ i ] < 1 )
        data[ i ] = 1;
    if( data[ i ] > 179 )
        data[ i ] = 179;
}
cout << "经换算后各舵机角度为：" << endl;
cout << "5/1:" ;                         //舵机号码
cout << data[ 0 ] << endl;               //将算出的舵机度数显示出
cout << "6/2:" ;
cout << data[ 1 ] << endl;
cout << "7/3:" ;
cout << data[ 2 ] << endl;
cout << "13/8:" ;
cout << data[ 3 ] << endl;
cout << "14/9:" ;
cout << data[ 4 ] << endl;
cout << "15/10:" ;
cout << data[ 5 ] << endl;
cout << "16/11:" ;
cout << data[ 6 ] << endl;
cout << "17/12:" ;

```

```

cout << data[7] << endl;
}
//////////////////////////////镜像法函数///////////////////////////

```

程序备注：

H1 [i] 代表 H1 的第 i 号舵机的转动角度；H2 [j] 代表 H2 的第 j 号舵机的转动角度。

上面所列的函数需运行两次才能得到对称动作的完整数据。例如：运行第一次时先利用对称动作前的 1、2、3、8、9、10、11、12 号舵机度数来算出对称动作后的 5、6、7、13、14、15、16、17 号舵机度数，再运行一次利用对称动作前的 5、6、7、13、14、15、16、17 号舵机度数算出对称动作后的 1、2、3、8、9、10、11、12 号舵机度数。

8) 在 main 函数里编写对称动作数据转化的角度计算，其中函数的重要部分就是利用式(7-3)的推论。在这里，我们已经给同学们一些已经存在的动作数据，这个数据是机器人招手的动作，但只是其中左手招手的动作，这里要求同学们利用函数得到右手招手的动作。若函数编写成功，我们立即将编写好的 VC 源代码直接生成出相应的工程，然后应该出现图 7-77 所示的界面。

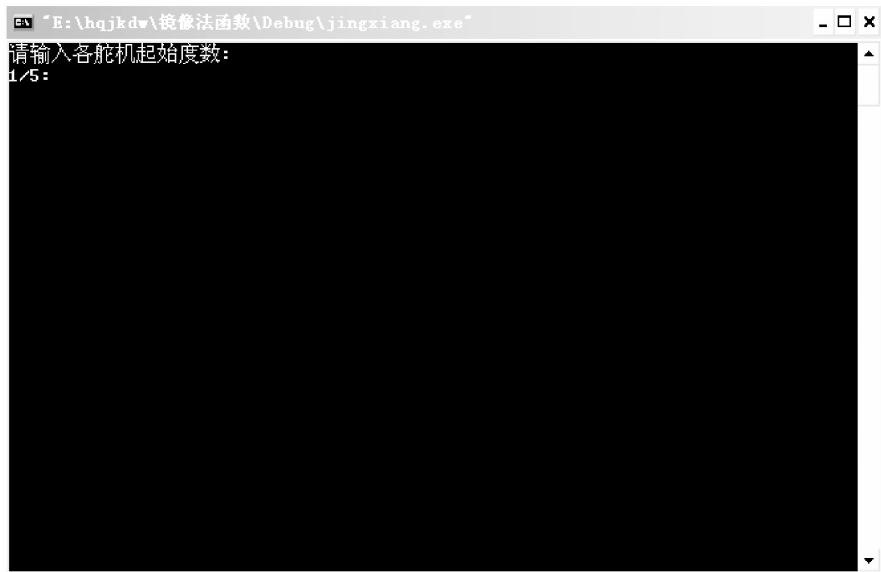


图 7-77 原像及镜像舵机角度转换界面

9) 在上位机软件中调出机器人左手招手动作，并在动作数据列表框当中显示出。把数据列表中的第 2 行 ch1、ch2、ch3、ch8、ch9、ch10、ch11、ch12 8 个舵机的数据，在图 7-77 所示界面“请输入各舵机起始度数：”部分中，填入原像舵机的起始度数，然后按回车，运行程序。其中，图 7-78 实例中的“1/5”代表这个初始位置既可以先规定为 1 号舵机的度数，也可以规定为 5 号舵机的度数，因此我们规定在界面中“请输入各舵机起始度数：”的部分，在“/”前面的数字为要输入起始角度的舵机号，对其输入角度值即可，直到全部 8 台原像舵机起始角度都输入完，才可输出结果，如图 7-78 界面中的“经换算后各舵机角度为：”部分。在此部分中“/”前面的数字号表示镜像舵机的转动角度值。直到全

部 8 台镜像舵机的角度值都输出显示为止。之后再运行一次程序，在界面中“请输入各舵机起始度数”的部分，输入“/”后面的数字号的舵机转动角度，对应地在界面中的“经换算后各舵机角度为：”部分，输出的结果也是对应“/”后面的数字号的舵机应当转动的角度。在此，由于动作 2 的数据另外 8 行都是保持在初始位置没动，因此就不用再通过镜像函数来计算了，因为即使计算了，得到的 8 行镜像数据组（由前八行的原像舵机角度计算出来后 8 行的镜像舵机角度，就称之为镜像数据组）也还都是初始位置。

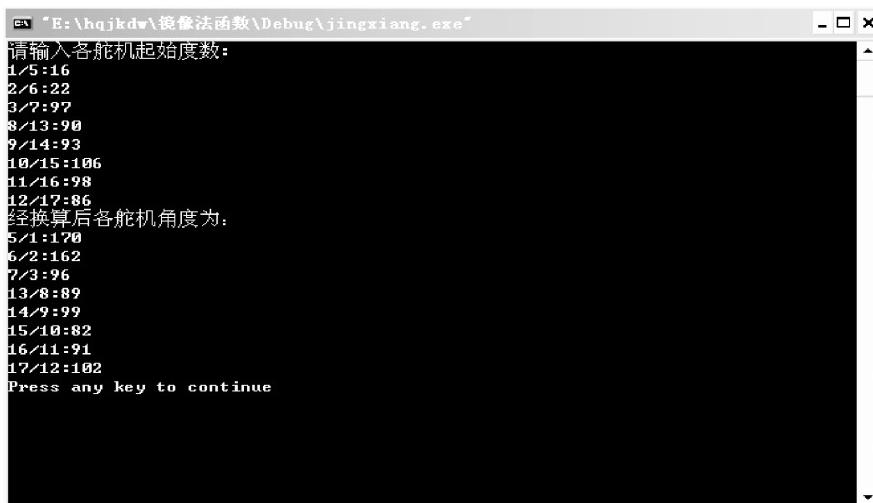


图 7-78 数据计算后显示的界面

10) 如图 7-78 所示，由前 8 行的原像舵机角度计算出来后 8 行的镜像舵机角度，并记录下这 8 个镜像舵机角度，称之为动作 2 的镜像数据组。再回到步骤 9 中数据列表框第 3 行，按照生成动作 2 的镜像数据组的方法，生成动作 3 的镜像数据组。通过观察上位机的动作数据文件，我们可以发现，以后的第 4 行和第 5 行以及第 6 行和第 7 行动作其实就是动作 2 和动作 3 的简单重复，因此没有必要再计算了。下面我们再双击上位机的第 8 行动作数据，然后单击添加按钮，连续单击 8 下，向动作数据文件再添加进去 8 行动作数据，这些数据每一行都是机器人的初始位置，下面我们来修改这些数据。首先双击第 10 行的动作数据，通过拉动 ch5、ch6、ch7、ch13、ch14、ch15、ch16、ch17 的角度条，把我们前面记下的动作 2 的镜像数据组改到相应的舵机上，然后单击修改，即完成了第 10 行动作数据的调整。再通过同样的方法把记下的动作 3 的镜像数据组修改到第 11 行动作数据中。根据前面招手动作采用的简单重复办法，我们分别把第 10 行和第 11 行的动作数据修改到第 12 行和第 13 行，以及第 14 行和第 15 行，最后一行保持初始位置不变。这时我们就得到一个完整的招手的数据，如图 7-79 所示。

11) 运行上位机程序，对机器人进行操作，若得到的数据正确，则我们能看到机器人能够成功的用右手向我们打招呼，这就是我们需要得到的动作，它正好是和机器人的招左手的动作是对称的，这样就达到了实验要求。

最后感兴趣的同学可以继续做下去，可以让机器人多实现几个对称的动作。



图 7-79 数据插入到上位机后的界面

五、思考题

1. 我们一共有几种方法可以得到双足步行机器人的动作数据？

答：(1) 最普通的方式，就是利用上位机里面的滚动条手动调出动作数据，该方法效率不高，但简单易行。

(2) 通过仿真来得到动作数据，这种方法完全抛开机器人本体，直接在虚拟的方针平台上对机器人进行数据调制，得到机器人的动作数据。

(3) 通过编程来得到动作数据，这种方法往往是当机器人的动作有取巧之处时，我们可以知道下一个动作与前面的某个动作有一定的对应关系，比如周期性、对称性等。该方法效率高，但有一定的难度。

(4) 通过行为数据库来得到机器人的动作数据，行为数据库里装满了机器人的各个动作的元素，合理的组合编排即可得出机器人的动作数据，该方法效率高，且简单易行，是现在比较流行的方法。

2. 除了上楼梯，还有那些动作可以利用镜像法来编排动作？

答：如果熟悉了镜像法的原理，不难发现只要编排的机器人动作里面有对称的关系，都可以利用镜像法来得到。比如：向左侧滚翻如果已经得到了，那么向右侧滚翻就可以用镜像法得到；还有有关前进时往前迈的左右腿都是对称的关系；还有一些舞蹈动作的编排，需要对称时，都可以利用镜像法来得到。

3. 除了镜像法，你还能总结出什么快捷简便的方法来得到数据？

答：(1) 周期法，当机器人的动作是周期性反复的话，就没有必要一帧一帧地编写，可以完全编写好一个完整周期后，即可利用周期法实现想要的动作。

(2) 倒序法，在机器人已有的动作前提下，我们可以利用倒序法来得到一些新的动作。例如：将机器人的前走，一帧一帧地倒写，就可以得到机器人的后走；将机器人向左侧翻，一帧一帧地倒写，就可以得到机器人的向右侧翻。

(3) 文件转化法，在机器人已有的动作的前提下，想得到另一个机器人的相同动作的数据，可以利用文件转化法来得到，但需要微调一下，计算往往会出现一两度的偏差。

(4) 临摹法，对于一些简单的动作且在机器人已有的动作的前提下，想得到另一个机器人的相同这样动作的数据，可以用临摹法，让已经有动作的机器人做“模特”，一帧一帧地摆动作，让其他机器人学习动作，达到实现动作的目的。

实验七 上位机编程实验

一、实验目的

- 1) 掌握 Visual C ++ 开发工具的基本使用方法。
- 2) 了解和掌握 Visual C ++ 工具编写基于 MFC Windows 应用程序的方法。
- 3) 熟悉和掌握几种控件的使用。

二、实验设备

PC 一台（装有 Visual C ++ ）。

三、实验原理与说明

在本实验中，我们将通过开发一个基于 MFC 的 Windows 应用程序，来了解 Visual C ++ 开发对话框程序的基本方法和步骤，为进一步开发机器人的上位机控制程序打好基础。这个程序的基本功能是：通过滚动条控制文本框显示的数值。

MFC 是微软开发的一个类库，它提供了开发 Windows 应用程序需要的基本框架，利用它可以极大地方便程序员开发 Windows 程序。MFC 以其强大的功能和灵活的编程方式成为大多数程序开发人员最经常使用的一种编程方式。我们使用的开发工具是 Visual C ++ 6.0，用这个微软提供的集成开发环境开发对话框类型的应用程序，只需要借助它提供的向导就可以方便地产生一个对话框程序的雏形，留给开发人员的工作就是各种控件的使用。例如在本实验的示例程序中大部分功能实际是在 MFC 的 CWinApp 和 CDialog 等基类中完成的，在编程时，只需在派生类中编写少量功能代码（这在下面的内容中会得到验证），

C++ 允许以这样的方式借用基类中的大量代码而无须复制代码。应用程序框架负责提供程序的结构框架，开发人员在此基础上为其添加相应的实现代码，从而可以非常方便地完成一个完整的应用程序。应用程序框架不仅定义了应用程序的结构安排，实际上还包含了更多的 C++ 基类。

本程序中用到的其他控件的使用，如文本框、滚动条等，可以参考相关的书籍，这方面的书籍很多，所以这里就不再说明了。

四、实验步骤

1. 建立一个基于对话框的工程

利用 Visual C++ 的向导建立一个基于对话框的应用程序，具体操作是：文件→新建，打开新建对话框，如图 7-80 所示。选择 MFC AppWizard (exe)，设置好工程名和路径，单击确定，就进入了应用程序向导，如图 7-81 所示，按照提示操作就生成对话框程序的基本框架。一般情况下，我们采用向导中的默认设置即可，不需要任何修改，所以直接单击“确定”及“完成”就生成了我们需要的工程。



图 7-80 新建工程对话框

2. 利用资源编辑器生成“上位机程序实验”界面

如图 7-82 所示，我们应该在窗体中添加这些控件：两个静态文本框，一个可编辑文本框，一个滚动条控件。完成这些操作也很简单，控件工具栏中选择相应的控件，然后按住鼠标左键拖拽即可完成摆放控件的工作。

3. 设置控件属性

在资源编辑器中，选中相应的控件，然后单击鼠标右键，选择“属性”，如图 7-83 所示。打开图 7-84 所示的属性设置对话框。

设置静态文本框的标题分别是：舵机角度和舵机角度控制；设置可编辑文本框的 ID 为



图 7-81 应用程序向导



图 7-82 程序界面

IDC_EDIT_DSHOW；设置滚动条的 ID 为 IDC_SCROLLBAR_DCTRL。其他属性不用修改，采用默认的设置即可。

4. 在类向导中添加变量和函数

通过菜单“查看→类向导”或者直接用快捷键“Ctrl + w”打开类向导，如图 7-85 所示。选择“Member Variables”标签，为可编辑文本框和滚动条添加变量，如图 7-86、图 7-87 所示。m-DShow 的类别选择 Value，变量类型选择 int；m-DCtrl 的类别选择 Control，变量类型选择 CScrollBar。在 MFC 中，变量与控件之间通过一定的映射机制关联，我们通过变量来操纵控件的行为。如我们通过 m_DShow 修改文本框控件显示的内容；通过 m_DCrl 控制滚动条的动作。



图 7-83 打开属性对话框的方法



a) 静态文本框



b) 可编辑文本框



c) 滚动条

图 7-84 各个控件的属性对话框

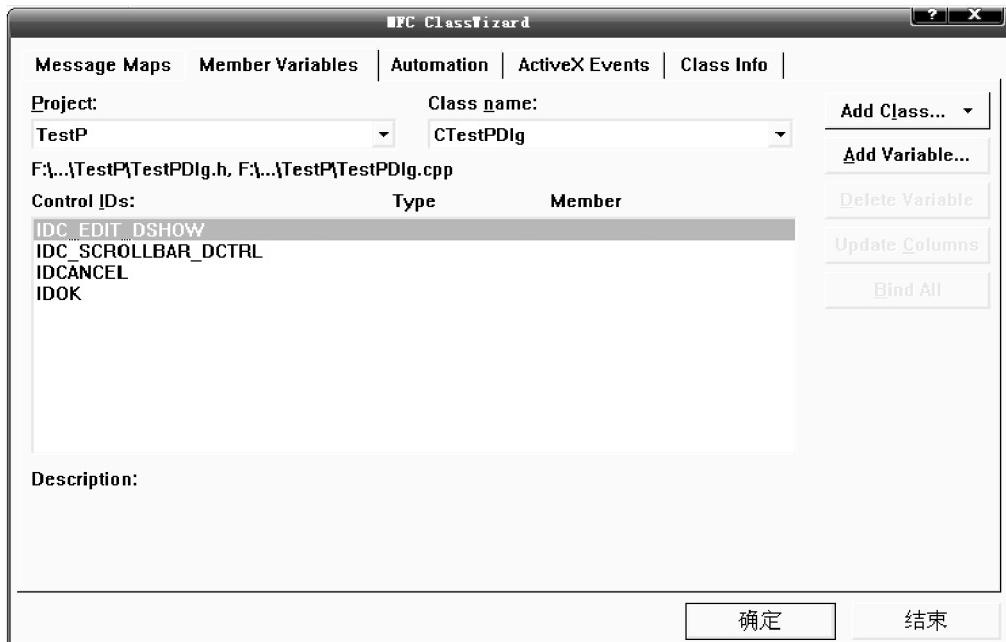


图 7-85 类向导对话框

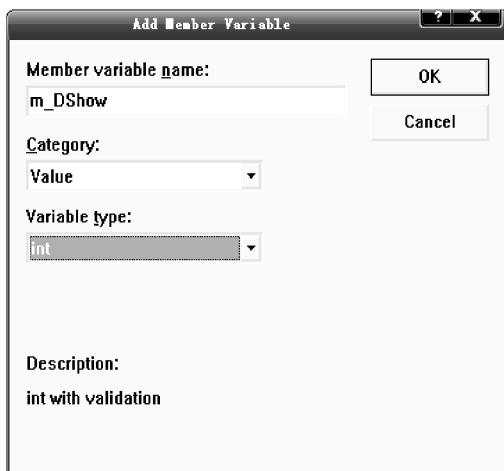


图 7-86 添加变量对话框

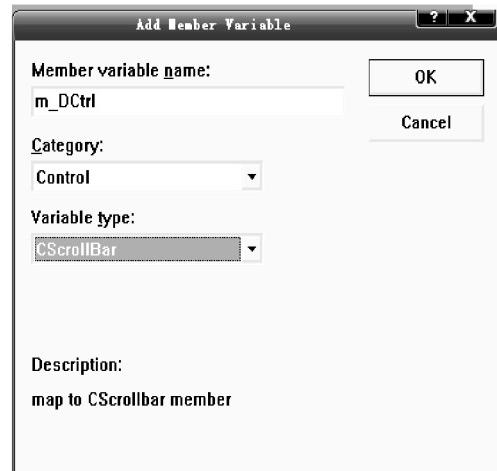


图 7-87 添加变量对话框

下面打开 Message Maps 标签，为主对话框的 WM_HSCROLL 消息添加函数，如图7-88 所示。这样，当用户拖动滚动条时，就会产生 WM_HSCROLL 消息，通过信息映射机制（MFC 已经自动为我们做好这个映射），程序就会调用 OnHScroll 函数。下一步我们将把控制文本框显示的代码放到 OnHScroll 函数中实现，就可以实现拖动滚动条控制文本框显示的目的了。单击“确定”完成所用的设置。

5. 添加自己的代码

找到 TestPDialog.cpp 文件中的 OnInitDialog 函数，在如图 7-89 所示的位置添加如下代码：

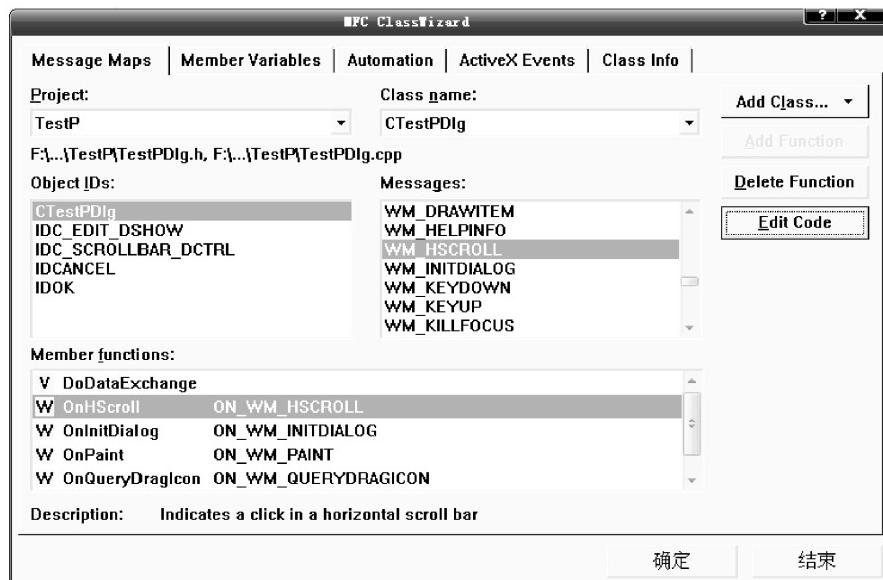


图 7-88 类向导对话框—添加消息函数

```

m _ DShow = 90;
m _ DCtrl. SetScrollRange( NMin ,NMax );
m _ DCtrl. SetScrollPos( ( NMin + NMax)/2,TRUE );
UpdateData( FALSE ); //使文本框显示为 m _ DShow 的值 90
SetIcon(m_hIcon, TRUE); // Set big icon
SetIcon(m_hIcon, FALSE); // Set small icon

// TODO: Add extra initialization here
|
return TRUE; // return TRUE unless you set the focus to a control

```

图 7-89 OnInitDialog 函数中位置

OnInitDialog 函数将在显示对话框之前调用，完成一些初始化工作。如上面的代码将使文本框显示为 90，滚动条位于中间位置。记住要给 NMin、NMax 进行声明，否则程序在编译的时候就会报错。

```
#define NMax 179
#define NMin 1
```

找到 TestPDIg.cpp 文件中的 OnHScroll 函数，在如图 7-90 所示的位置添加如下代码：

```

switch( nSBCode )
{
    case SB - THUMBTRACK:
        if( ( * pScrollBar ) == m - DCtrl ) {
            m - DShow = nPos; ( * pScrollBar ). SetScrollPos( m - DShow );break; }
        break;
    case SB _ LINERIGHT:

```

```

case SB_PAGERIGHT:
    if( (* pScrollBar) == m_DCtrl) {
        if( m_DShow ++ >= NMax)
            m_DShow = NMax; (* pScrollBar). SetScrollPos( m_DShow);
        break;
    }
break;

case SB_LINELEFT:
case SB_PAGELEFT:
    if( (* pScrollBar) == m_DCtrl) {
        if( m_DShow -- <= NMin)
            m_DShow = NMin; (* pScrollBar). SetScrollPos( m_DShow);
        break;
    }
break;
}

UpdateData( false);

void CTestPDLg::OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar)
{
    // TODO: Add your message handler code here and/or call default
    CDIalog::OnHScroll(nSBCode, nPos, pScrollBar);
}

```

图 7-90 OnHScroll 函数中位置

6. 编译、构建并执行程序

通过菜单：编译→构建，建立工程；编译→执行，执行工程。或者通过工具栏进行这些操作，如图 7-91 所示。



图 7-91 编译工具栏

程序运行后的界面如图 7-92 所示，同学们可以拖动滚动条看看文本框内容的变化。

五、思考题

1. 用自己的话概述消息处理的过程？

答：在 Window 操作环境下，无论是系统产生的行为或用户运行应用程序产生的行为都称事件产生的消息，消息处理函数处理消息的过程一般归纳为接收消息、分发消息、处理消息和用户交互。



图 7-92 程序界面

2. 总结 OnHScroll 函数的用法?

答：当用户拖动滚动条时，就会产生 WM_HSCROLL 消息，通过信息映射机制（MFC 已经自动为我们做好这个映射），程序就会调用 OnHScroll 函数。

实验八 双足步行机器人串行口通信实验

一、实验目的

- 1) 了解串口通信的基本原理。
- 2) 了解双足步行机器人读取初始位置时的通信协议格式。
- 3) 了解 VC 串口通信控件的应用。
- 4) 学习机器人的上位机和下位机是如何实现交互从而控制机器人的。

二、实验设备

有串口的 PC 一台（装有 VC 编程环境及机器人上位机软件）、双足机器人一台、串口通信线一条。

三、实验原理与说明

串口通信的基本原理及 VC 串口编程：

1. 串行通信的基本原理

所谓“串行通信”是指外设和计算机间使用一根数据信号线，数据在一根数据信号线上按位进行传输，每一位数据都占据一个固定的时间长度。当数据从 CPU 经过串行端口发送出去时，字节数据转换为串行的二进制位。在接收数据时，串行的二进制位被转换为字节数据。在 Windows 环境（Windows NT、Win98、Windows2000）下，串口是系统资源的一部分。应用程序要使用串口进行通信，必须在使用之前向操作系统提出资源申请要求（打开串口），通信完成后必须释放资源（关闭串口）。

2. 串口信号线的接法

一个完整的 RS-232C 接口有 22 根线，采用标准的 25 芯插头座（或者 9 芯插头座）。25

芯和 9 芯的主要信号线相同。以下的介绍是以 25 芯的 RS - 232C 为例。

(1) 主要信号线定义

- 2 脚：发送数据 TXD；
- 3 脚：接收数据 RXD；
- 4 脚：请求发送 RTS；
- 5 脚：清除发送 CTS；
- 6 脚：数据设备就绪 DSR；
- 20 脚：数据终端就绪 DTR；
- 8 脚：数据载波检测 DCD；
- 1 脚：保护地；
- 7 脚：信号地。

(2) 电气特性 数据传输速率最大可到 20kbit/s，最大距离仅 15m。

(3) 接口的典型应用 大多数计算机应用系统与智能单元之间只需使用 3 ~ 5 根信号线即可工作。这时，除了 TXD、RXD 以外，还需使用 RTS、CTS、DCD、DTR、DSR 等信号线（当然，在程序中也需要对相应的信号线进行设置）。

以上接法，在设计程序时，直接进行数据的接收和发送就可以了，不需要对信号线的状态进行判断或设置（如果应用的场合需要使用握手信号等，需要对相应的信号线的状态进行监测或设置）。

3. 在 MFC 下的 32 位串口应用程序

32 位串口通信程序可以用两种方法实现：利用 ActiveX 控件；使用 API 通信函数。使用 ActiveX 控件，程序实现非常简单，结构清晰，缺点是欠灵活；使用 API 通信函数的优缺点则基本上相反。

在此，我们是用 MSComm 控件实现。我们上位机所要作的工作是利用 VC 里的 MSComm 控件的方法来进行双向的单线程串口通信。

这牵扯到数据协议的制定和编程的实现。上位机与下位机的通信方式有有线和无线两种，有线方式已经比较成熟，无线方式需要对单向的传输增加握手协议，使下位机可以向上位机反馈信息。由于不管是采用有线还是无线方式，上位机都是通过串口与下位机通信，所以编程部分只需要根据串口协议的变化做出相应的调整就可以了。

四、实验步骤

1. 了解双足机器人的通信协议格式

数据包格式：0xff, 0xfe + 数据帧 + 校验和。

数据帧格式：地址码 + 控制命令 + 0x00, …, 0x00。

控制命令：0x01：主机读回 Homeposition 数据。

机器人返回数据：地址码 + 数据标签 + 相关参数（4 字节）+ Homeposition（17 字节）。

数据标签：0x00：Homeposition 数据（主机读 Homeposition 时）。

2. 编写串口监视软件

1) 创建工程，单击 File→New，产生如图 7-93 所示窗口，在左边选择“MFC AppWizard [exe]”，右面“Project name”处填写项目名称为：CommShow，选择保存路径，单击下一步。

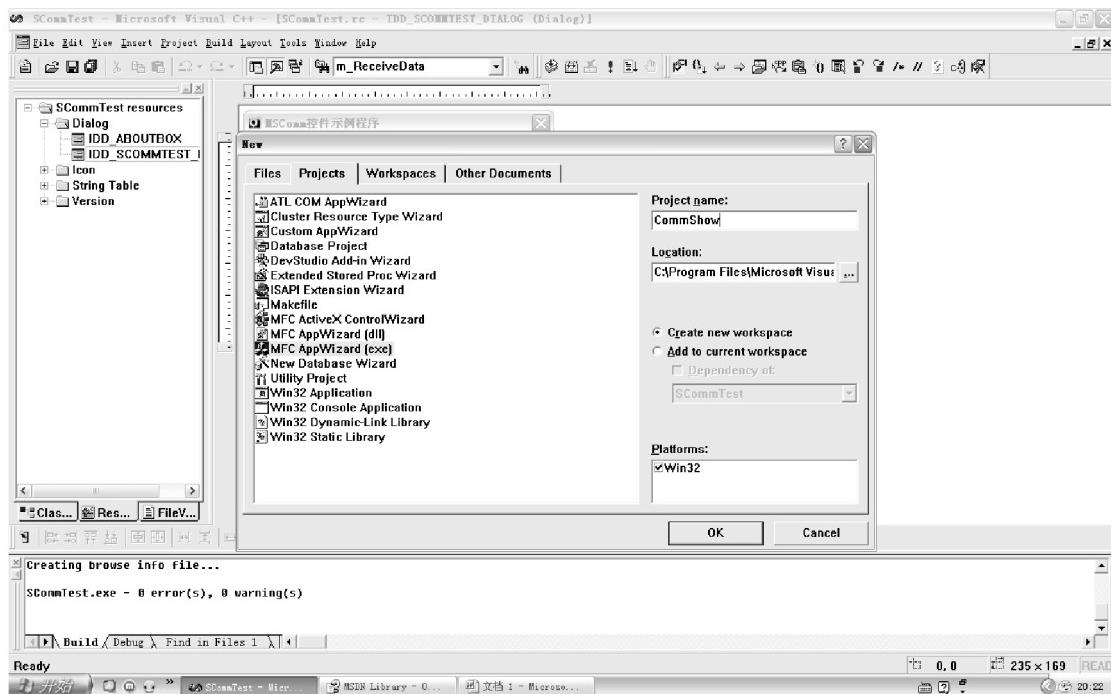


图 7-93 工程设置

2) 将项目设置为 Dialog based, 如图 7-94 所示, 然后单击完成。

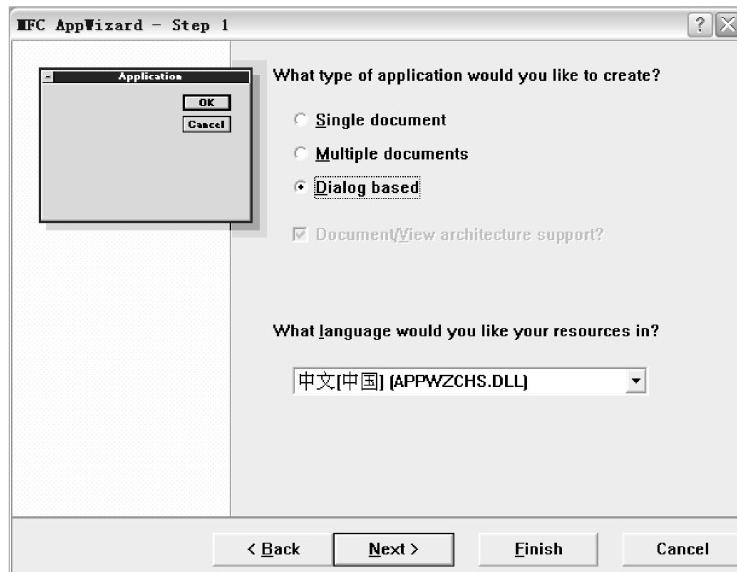


图 7-94 MFC 应用程序向导步骤

3) 在 Dialog 资源中加入如下控件: 添加 Static box 和 EditBox 控件各两个, Botton 控件一个, 并通过在控件上单击右键, 选择“Properties”进行控件属性设置, 改变 Staticbox 的 Caption 为“发送显示”和“接收显示”; 将 Botton 控件的 Caption 改为“发送”; EditBox 控件处

设置它们的 ID 分别为 IDC – EDIT1 和 IDC – EDIT2，并在属性窗口的“Style”选项卡中，作如图 7-95 所示设置。

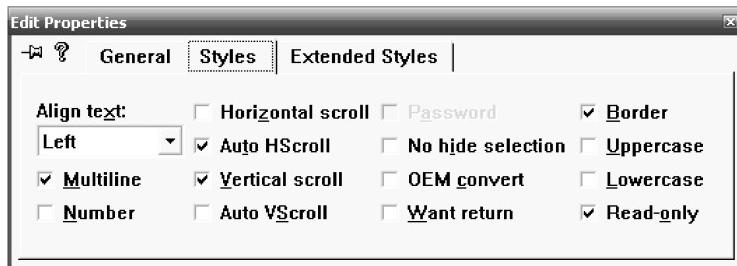


图 7-95 EditBox 控件属性框

全部设置完成后如图 7-96 所示。

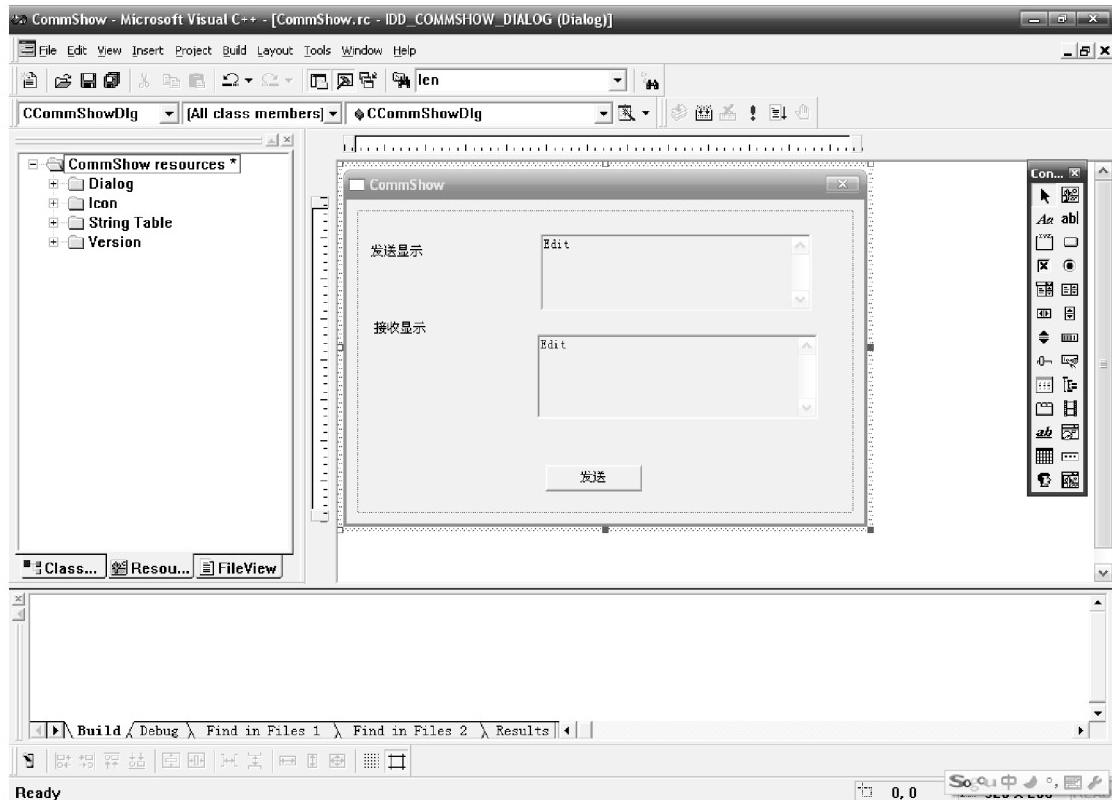


图 7-96 控件设置完成后界面

- 4) 单击 Project→Add To Project→Components and Controls，准备添加串口通信控件。
- 5) 查找范围选择 Registered ActiveX Controls，在 Registered ActiveX Controls 文件夹下，找到 Microsoft Communications Control 控件，单击“Insert”按钮如图 7-97 所示，弹出如图所示对话框，单击“OK”，之后单击“关闭”。
- 6) 将通信控件 拖动，添加到对话框中，并单击右键，选择 Properties 项，设置如图 7-98 所示。

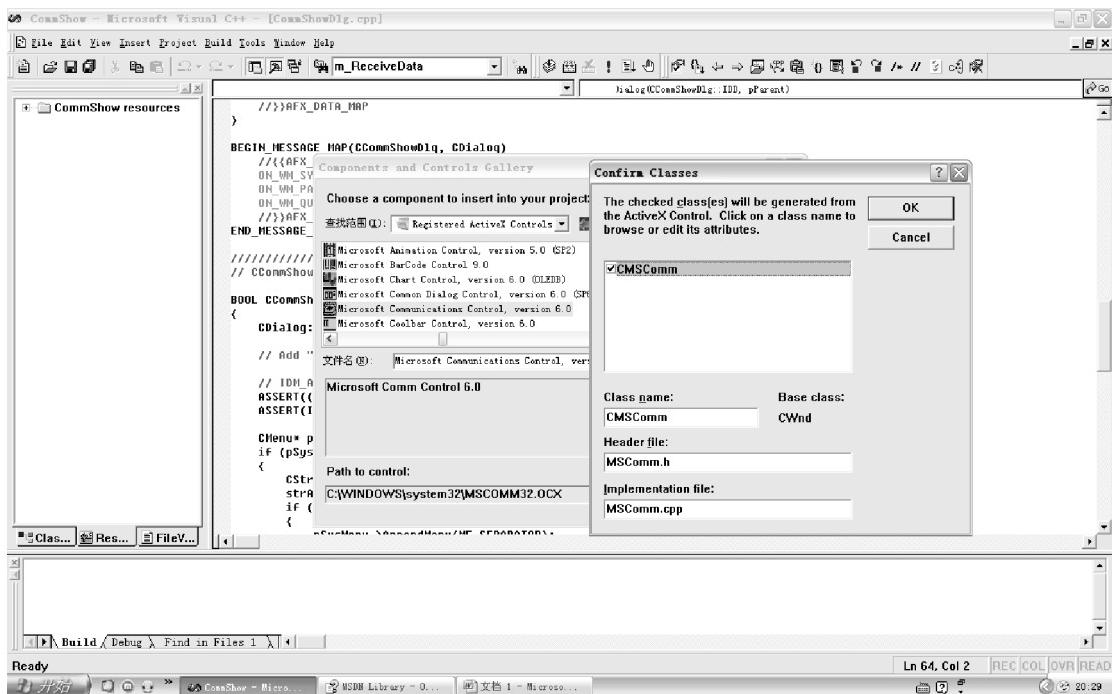


图 7-97 添加串口通信控件到基本工具栏中

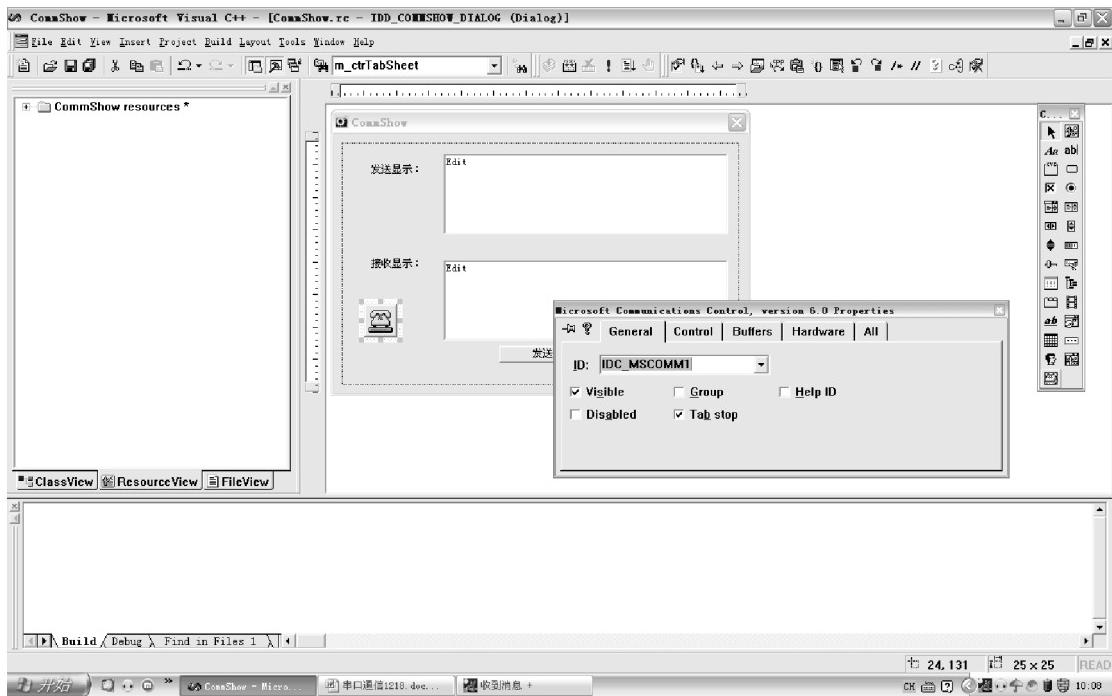


图 7-98 设置串口通信控件 ID 号

7) 在 Dialog 资源中单击右键，选择 ClassWizard，为各资源加入变量如图 7-99 所示。

通过“MFC ClassWizard”对话框上面的选项卡，选择 Member Variables，通过双击控件的 ID 弹出 Add Member Variable 对话框来添加控件的变量，如图 7-100 所示。

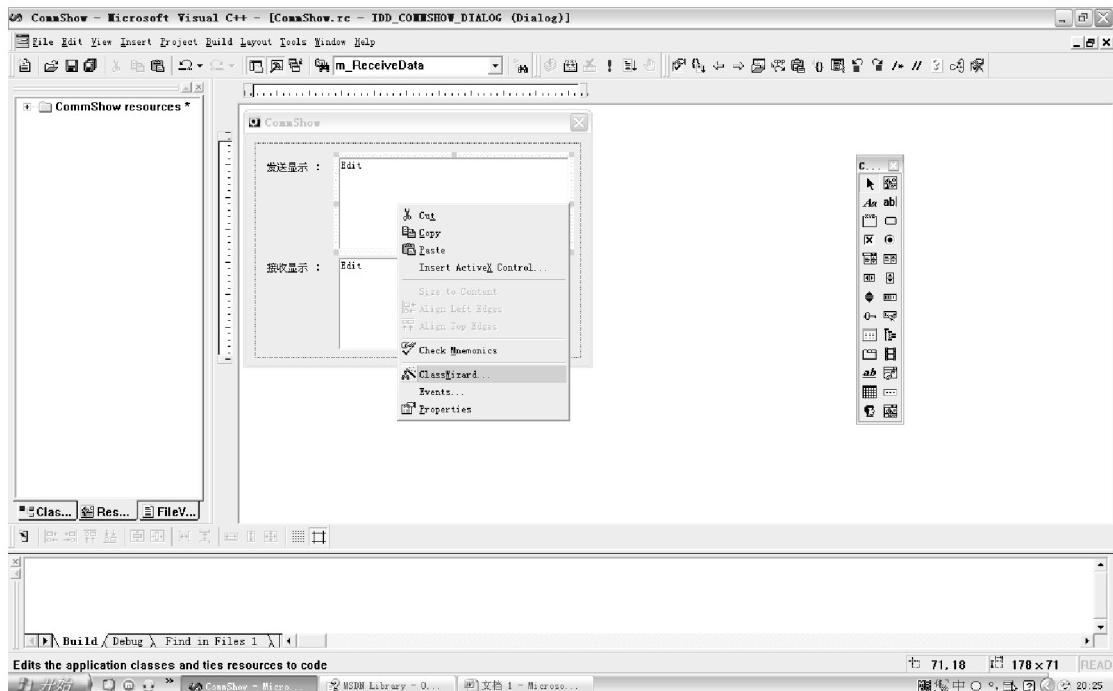


图 7-99 选择 ClassWizard 进入类向导

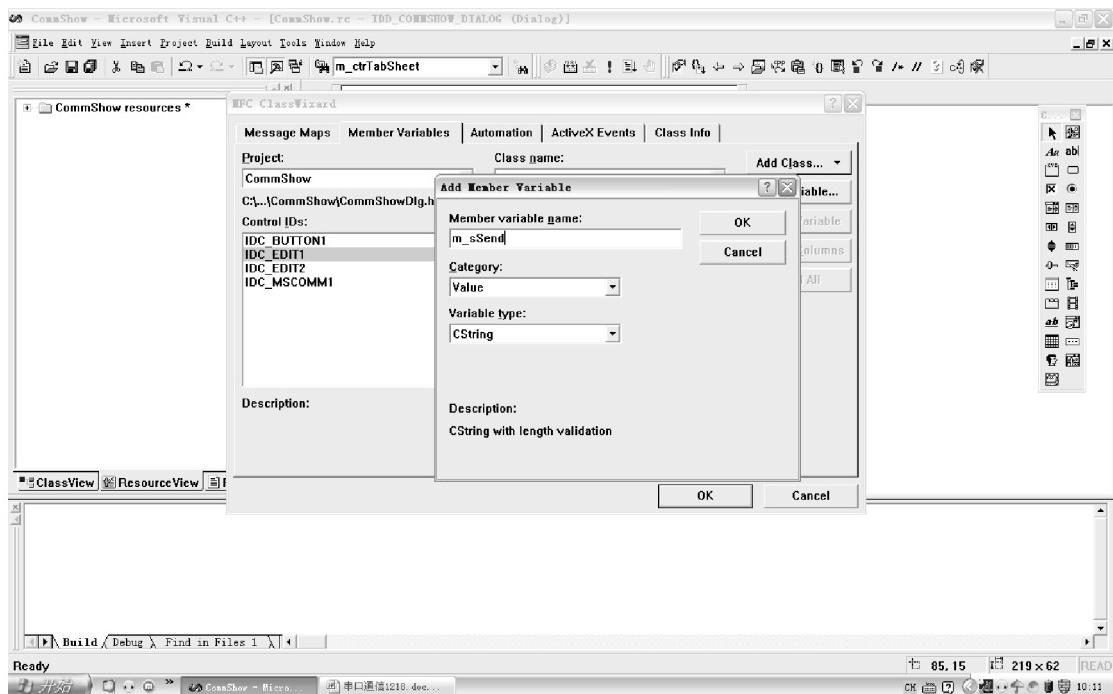


图 7-100 为各控件添加成员变量

添加后的结果如图 7-101 所示，之后单击“OK”退出。

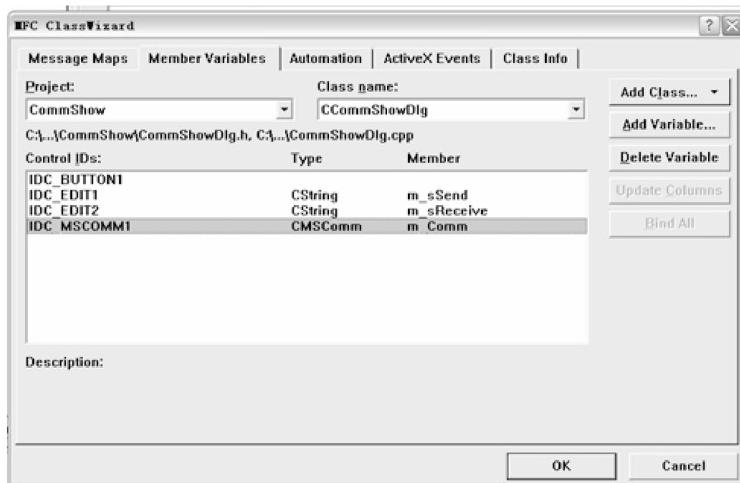


图 7-101 完成添加全部成员变量后界面

8) 在 MFC ClassWizard 界面中，选择 Message Maps 选项卡，单击 IDC – MSCOMM1 和 Messages 框中的 OnComm，当两条都被选中时，单击右面的“Add Function”按钮，准备加入消息响应函数 OnComm ()，如图 7-102 所示。

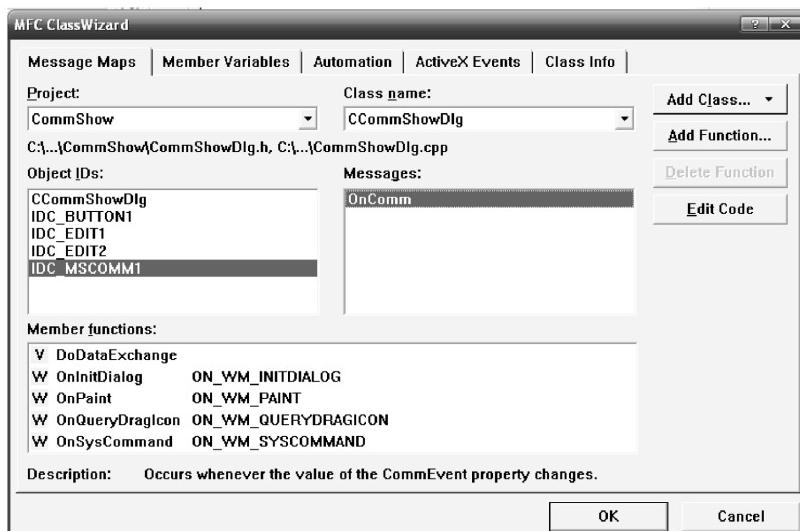


图 7-102 准备添加消息响应函数 OnComm ()

弹出如图 7-103 所示对话框，进行如下设置后，单击“OK”按钮。



图 7-103 设置成员函数名

9) 双击“Member functions”框中的 OnComm 函数（见图 7-104），在弹出的 CommShowDlg.cpp 文档（见图 7-105）的 OnComm 函数中加入如下代码。

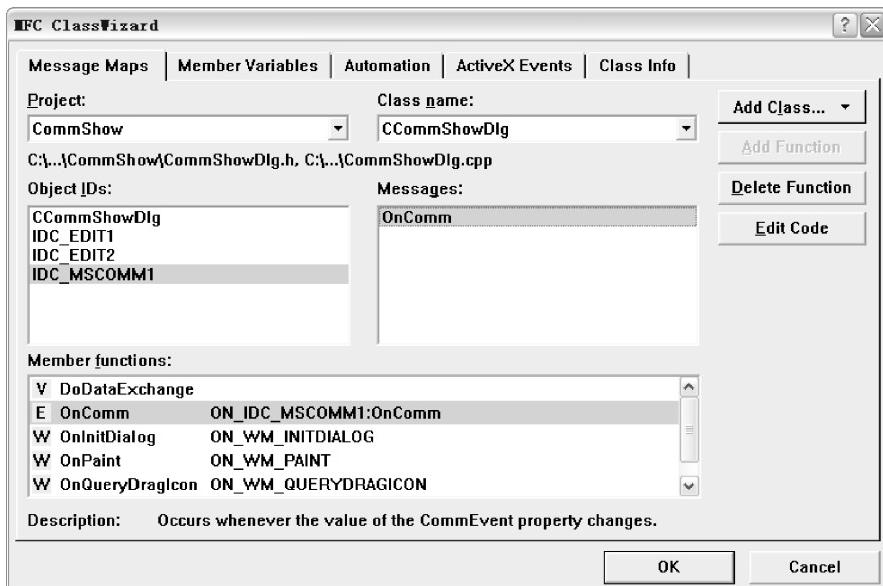


图 7-104 成员函数 OnComm () 设置完成

```

// Draw the icon
dc.DrawIcon(x, y, m_hIcon);
}
else
{
    CDialog::OnPaint();
}
}

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CCommShowDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

BEGIN_EVENTSINK_MAP(CCommShowDlg, CDialog)
//{{AFX_EVENTSINK_MAP(CCommShowDlg)
ON_EVENT(CCommShowDlg, IDC_MSCOMM1, 1 /* OnComm */, OnComm, UTS_NONE)
//}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()

void CCommShowDlg::OnComm()
{
    // TODO: Add your control notification handler code here
}

```

图 7-105 弹出的 CommShowDlg.cpp 文档

```

void CCommShowDlg::OnComm()
{
    // TODO: Add your control notification handler code here
    VARIANT variant_inp;

```

```

COleSafeArray safearray _ inp;
LONG len,k;
BYTE rxdata[ 2048 ] ; //设置 BYTE 数组
CString strtemp;

if( m _ Comm. GetCommEvent( ) == 2 )
{
    //响应串口接收事件的代码
    variant _ inp = m _ Comm. GetInput( ) ; //读缓冲区
    safearray _ inp = variant _ inp; //VARIANT 型变量转换为 ColeSafeArray 型变量
    len = safearray _ inp. GetOneDimSize( ) ; //得到有效数据长度
    for( k = 0 ; k < len ; k ++ )
        safearray _ inp. GetElement( &k ,rxdata + k );//转换为 BYTE 型数组
    for( k = 0 ; k < len ; k ++ ) //将数组转换为 Cstring 型变量
    {
        BYTE bt = * ( char * ) ( rxdata + k ); //字符型
        strtemp. Format( "% x " ,bt); //将字符送入临时变量 strtemp 存放
        m _ sReceive += strtemp; //加入接收编辑框对应字符串
    }
    m _ sReceive += "\r\n";
    UpdateData( FALSE ); //更新编辑框内容
}
}

```

10) 在 BOOL CCommShowDlg:: OnInitDialog () 函数的 // TODO: Add extra initialization here 代码后加入程序初始化代码，进行串口初始化。

```

// TODO: Add extra initialization here
// 初始化串口
m _ Comm. SetCommPort(1); //选择 COM1
m _ Comm. SetInputMode(1); //输入方式为二进制方式
m _ Comm. SetInBufferSize(1024); //设置输入缓冲区大小
m _ Comm. SetOutBufferSize(512); //设置输出缓冲区大小
//波特率 9600,无校验,8 个数据位,1 个停止位
m _ Comm. SetSettings( "9600,n,8,1" );
if( ! m _ Comm. GetPortOpen() )
    m _ Comm. SetPortOpen( TRUE ); //打开串口
//参数 1 表示每当串口接收缓冲区中有多于
//或等于 16 个字符时将引发一个接收数据的 OnComm 事件
m _ Comm. SetRThreshold(25);
m _ Comm. SetInputLen(0); //设置当前接收区数据长度为 0

```

```
m _ Comm. GetInput( ); //先预读缓冲区以清除残留数据
//////////////////////////////
```

11) 双击“发送”按钮，添加发送函数：双击后，在弹出的对话框中单击“OK”按钮，之后进入函数内加入代码。

```
void CCommShowDlg::OnButton1()
```

```
{
```

```
    unsigned char chTemp[26];
    chTemp[0] = 0xff;
    chTemp[1] = 0xfe;
    chTemp[2] = 127;
    chTemp[3] = 0x01;           //读 Homeposition
    for( int i = 4; i <= 24; i ++ )
        chTemp[ i ] = 0x00;
```

//校验和

```
unsigned char sum = 0;
for ( i = 0; i < 25; i ++ )
{
    sum + = chTemp[ i ];
}
```

```
chTemp[25] = - sum;
```

//通过串口发送数据

```
CByteArray array;           //定义转换的数据类型
array. RemoveAll();         //清空 array
array. SetSize(26);         //设定 array 的大小
```

```
for( i = 0; i < 26; i ++ )
```

```
{
```

```
    array. SetAt(i, * ( chTemp + i )); //给 array 赋值
```

```
    BYTE bt = * ( char * )( chTemp + i ); //字符型
```

```
    CString strtemp;
```

```
    strtemp. Format( "% x ", bt ); //将字符送入临时变量 strtemp 存放
```

```
    m _ sSend + = strtemp; //加入接收编辑框对应字符串
```

```
}
```

```
    m _ sSend + = "\r\n" ;
```

```
    m _ Comm. SetOutput( COleVariant( array ) );
```

```
    UpdateData( FALSE );
```

```
}
```

12) 通过依次单击菜单或工具栏中  和  按钮，进行编译并执行程序。将程序编译并执行之后界面如图 7-106 所示。

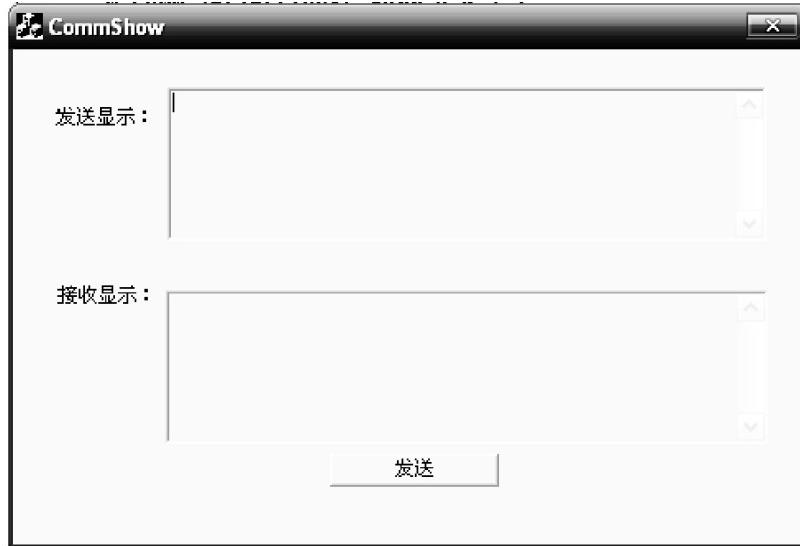


图 7-106 程序编译执行结果

3. 连接好机器人

采用有线方式，机器人背部电路板上两个模式拨码开关均拨到下方，串口选择 COM1 控制，单击“发送”按钮。利用实验步骤 1 中的预备知识并通过软件观察数据的格式，了解上位机和下位机具体是如何利用串口进行通信的。

- 1) 打开编写的软件。
 - 2) 单击“发送”按钮，通过软件观察数据，如图 7-107 所示。



图 7-107 程序运行结果

五、思考题

1. 回顾实验过程中，上下位机的通信过程，总结上位机是如何通过串口与机器人实现通信的？

答：通过软件上下的显示，可以看出上位机先发出了机器人控制信息，如：机器人工作状态、下位机存储的动作数据组号、数据帧、舵机角度及校验等信息。下位机在收到数据后验证校验和，并且给上位机相应的反馈。由此，保证上下位机之间正确、可靠的通信。

2. 思考使用多线程进行上下位机同步通信的意义？

答：多线程提高了程序执行的效率，提高了编程的灵活性。在串口中程序中，有很多应用多线程设计的地方，例如：为了实现串口接收的事件驱动方式，可以考虑单独建立一个工作线程。在该线程中循环检测串口事件，但是不会造成程序界面的假死；在使用多个串口时，也可以使用多线程编程。

实验九 ZMP 实验

一、实验目的

- 1) 掌握杆件位姿矩阵的建立方法。
- 2) 了解 ZMP 理论及其在机器人控制中的重要意义。
- 3) 了解使用 MATLAB 对 ZMP 理论的验证做出仿真。

二、实验设备

双足步行机器人一台、PC 一台（装有 MATLAB，上位机控制软件）。

三、实验原理与说明

1. ZMP 点计算及步行稳定条件

双足步行机器人研究的一个关键问题是实现其稳定的行走。目前世界上大多数双足步行机器人系统都采用 ZMP 作为稳定行走的判据。ZMP 即零力矩点（Zero Moment Point）的简称，是由南斯拉夫学者 Vukobratovic（简称伍氏）提出的关于步行稳定性的经典理论，具体描述为：指重力与惯性力的合力的延线与地面的交点，在该点处，合力对 x 轴和 y 轴的力矩为零，如图 7-108 中合力 F 与地面的焦点就是 ZMP 点。这个 ZMP 点的位置直接影响到了机器人的稳定性，即机器人是否能够稳定的站立。

双足步行机器人在步行过程中机体的 ZMP 点是随时变化的，我们将 ZMP 离支撑面边缘的最短距离定义为稳定裕度，当在其内时为正，是稳定的，反之为负，是不稳定的。

步行机器人在步行时，作用在其上的外力为重力、摩擦

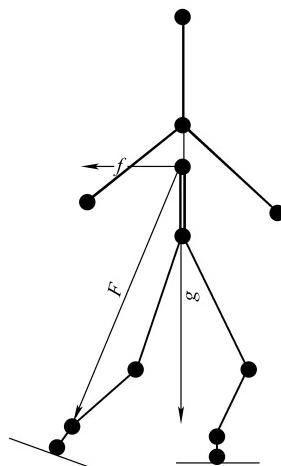


图 7-108 机器人受力分析与 ZMP 点的确定

力和惯性力。按行走的动态平衡理论, ZMP 点的规划直接影响到步行机器人的稳定性、补偿运动的大小、关节驱动力矩的大小, 所以在规划时应在保证动态平衡、行走稳定的前提下, 使补偿运动、关节驱动力矩尽量减小, 从而减小步行机器人的能量消耗, 提高步行效率。

根据 Vukobratov 提出的 ZMP 理论, 当支撑脚与地面多点接触时, ZMP 点的位置坐标为

$$X_{\text{ZMP}} = \frac{\sum_{i=0}^n m_i (\overset{00}{Z}_i + g) X_i - \sum_{i=0}^n m_i (\overset{00}{X}_i + g) Z_i}{\sum_{i=0}^n m_i (\overset{00}{Z}_i + g)} \quad (7-4)$$

$$Y_{\text{ZMP}} = \frac{\sum_{i=0}^n m_i (\overset{00}{Z}_i + g) Y_i - \sum_{i=0}^n m_i (\overset{00}{Y}_i + g) Z_i}{\sum_{i=0}^n m_i (\overset{00}{Z}_i + g)} \quad (7-5)$$

式中 m_i ——各部分的质量;

X_i 、 Y_i 、 Z_i ——各部分的质心。

当机器人做静态步行时, $\overset{00}{Z}_i = \overset{00}{X}_i = \overset{00}{Y}_i = 0$

所以有:

$$\begin{cases} X_{\text{ZMP}} = X_c = \sum_{i=1}^n m_i X_i / \sum_{i=1}^n m_i \\ Y_{\text{ZMP}} = Y_c = \sum_{i=1}^n m_i Y_i / \sum_{i=1}^n m_i \end{cases} \quad (7-6)$$

式中 X_c 、 Y_c ——双足机器人质心的坐标。

即, 双足机器人的质心即为 ZMP 点。因此只要使重心落在脚面上, 就可保证双足步行机器人稳定的静态步行。因此复杂的 ZMP 点求解的任务变成了求解重心点的任务。

2. 建立连杆坐标系, 确定位姿矩阵

双足步行机器人为多杆系统, 两杆间的位姿矩阵是求解双足步行机器人各个质点位姿矩阵的基础。两杆间的位姿矩阵取决于两杆之间的结构参数、运动形式和运动参数, 以及这些参数按照不同顺序建立的几何模型。

为了描述每个连杆与相邻连杆之间的相对位置关系, 需要在每个连杆上定义一个固连坐标系。根据固连坐标系所在连杆的编号对固连坐标系命名, 因此固连在连杆 i 上的固连坐标系称为坐标系 $\{i\}$ 。

通常按照下面的方法确定连杆上的固连坐标系: 坐标系 $\{i\}$ 的 Z 轴称为 Z_i , 并与关节轴 i 重合, 坐标系 $\{i\}$ 的原点位于公垂线 a_i 与关节轴 i 的交点处。 X_i 沿 a_i 方向由关节 i 指向关节 $i+1$ 。

当 $a_i = 0$ 时, X_i 垂直于 Z_i 和 Z_{i+1} 所在的平面。按右手定则绕 X_i 轴的转角定义为 α_i , 由于 X_i 轴的方向可以有两种选择, 因此 α_i 的符号也有两种选择。 Y_i 轴由右手定则确定, 从而完成了对坐标系 $\{i\}$ 的定义。图 7-109 所示为一般连杆坐标系 $\{i-1\}$ 和 $\{i\}$ 的位置。

在研究机器人运动学问题时, 我们都会选取一个参考坐标系, 并在找个参考坐标系中描述所有其他连杆坐标系的位置。

参考坐标系 $\{0\}$ 可以任意设定, 但是为了使问题简化, 通常设定 Z_0 轴沿关节轴 1 的

方向，并且当关节变量 1 为 0 时，设定参考坐标系 $\{0\}$ 与坐标系 $\{1\}$ 重合。

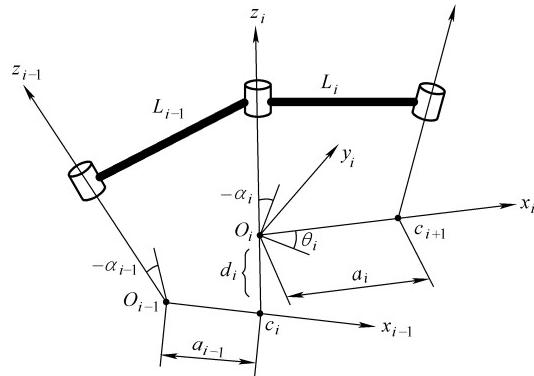


图 7-109 空间相邻的两相邻杆件

如果按照上述规定，则连杆参数规定如下：

α_{i-1} = 沿 x_{i-1} 轴，从 z_{i-1} 旋转到 z_i 的角度，记作 $\text{Rot}(x_{i-1}, \alpha_{i-1})$ ；

a_{i-1} = 绕 x_{i-1} 轴，从 z_{i-1} 移动到 z_i 的距离， $\text{Trans}(x_{i-1}, a_{i-1})$ ；

d_i = 沿 z_{i-1} 轴，从 x_{i-1} 移动到 x_i 的距离， $\text{Trans}(z_i, d_i)$ ；

θ_i = 沿 z_{i-1} 轴，从 x_{i-1} 旋转到 x_i 的角度， $\text{Rot}(z_i, \theta_i)$ 。

旋转 (α_{i-1}) —— 平移 (α_{i-1}) —— 平移 (d_i) —— 旋转 (θ_i)，变换矩阵是：

$T_i^{-1} = \text{Rot}(x_{i-1}, \alpha_{i-1}) \text{ Trans}(z_{i-1}, \alpha_{i-1}) \text{ Trans}(z_i, d_i) \text{ Rot}(z_i, \theta_i)$ 即：

$$\begin{aligned}
 T_i^{-1} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & \alpha_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &\quad \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ c\alpha_{i-1}s\theta_i & c\alpha_{i-1}c\theta_i & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\alpha_{i-1}s\theta_i & s\alpha_{i-1}c\theta_i & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7-7}
 \end{aligned}$$

确定位姿矩阵参数的步骤如下：

1) 找出各关节轴，并标出（或画出）这些轴线的延长线。在下面的步骤 2 到 5 中，仅考虑两个相邻的轴线（关节轴 i 和 $i+1$ ）。

2) 找出关节轴 i 和 $i+1$ 之间的公垂线或关节轴 i 和 $i+1$ 的交点，以 $i+1$ 之间的交点或公垂线与关节轴的交点作为连杆坐标系 $\{i\}$ 的原点。

- 3) 规定 Z_i 轴沿关节轴 i 的指向。
- 4) 规定 X_i 轴沿公垂线的指向, 如果关节轴 i 和 $i+1$ 相交, 则规定 X_i 轴垂直于关节轴 i 和 $i+1$ 所在的平面。
- 5) 按照右手定则确定 Y_i 轴。
- 6) 当第一个关节变量为 0 时, 规定坐标系 $\{0\}$ 和 $\{1\}$ 重合。对于坐标系 $\{N\}$, 其原点的方向可以任意选取。但是在选取时, 通常尽量使连杆参数为 0。

注意: 按照上述方法建立的连杆固连坐标系并不是唯一的。首先, 当选取 Z_i 轴与关节 i 重合时, Z_i 轴的指向有两种选择。此外, 在关节轴相交的情况下 (这时 $a_i = 0$), 由于 X_i 轴垂直与 Z_i 轴与 Z_{i+1} 轴所在的平面, 因此 X_i 轴的指向也有两种选择。当关节轴 i 与 $i+1$ 平行时, 坐标系 $\{i\}$ 的原点位置可以任意选择 (通常选取该原点使之满足 $d_i = 0$)。

3. 重心坐标的确定

根据机器人各个关节的质量和它的位姿矩阵 (在此不做推导), 我们可以求出机器人各个关节质点相对于坐标原点 (头部) 的坐标, 并且可进一步求出机器人的重心坐标。重心坐标的求解公式如下:

$$O_z = \left(\sum_i O_i \times m_i \right) / \sum_i m_i \quad (7-8)$$

式中 O_z ——重心的位置;

O_i ——第 i 个舵机质点的位置;

m_i ——第 i 个舵机质点的质量。

$$T_A^0 = \begin{bmatrix} R_A^0 & P_{OA}^0 \\ \hline 0 & 1 \end{bmatrix} \quad (7-9)$$

$$P_{OA}^0 = (O_0 O_A)^0 = \begin{bmatrix} x_{O_0 A} \\ y_{O_0 A} \\ z_{O_0 A} \end{bmatrix} = \begin{bmatrix} P_{x_o} \\ P_{y_o} \\ P_{z_o} \end{bmatrix} \quad (7-10)$$

式中 P_{OA}^0 —— O_A 点的径向 P_{OA} 在坐标系 S_0 中的表示, 是一个 3×1 列阵 (位置矩阵);

R_A^0 ——坐标系 S_A 在坐标系 S_0 中的表示, 是一个 3×3 的方阵 (姿势矩阵);

T_A^0 ——称为位姿矩阵, 用 4×4 的方阵同时把位置和姿态表示出来。

四、实验步骤

1. 简化连杆模型的坐标系建立

- 1) 根据连杆模型 I (见图 7-110), 将参数值填入表 7-9 中, 并写出其对应的位姿矩阵。

表 7-9 连杆模型 I 参数值表

| α_{i-1} | a_{i-1} | d_i | θ_i |
|----------------|-----------|-------|------------|
| | | | |

将其参数值带入式 (7-7) 中:

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ c\alpha_{i-1}s\theta_i & c\alpha_{i-1}c\theta_i & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\alpha_{i-1}s\theta_i & s\alpha_{i-1}c\theta_i & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

带入结果为

$$T_i^{i-1} = \begin{bmatrix} () & () & () & () \\ () & () & () & () \\ () & () & () & () \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2) 连杆模型 II (见图 7-111) 将参数值填入表 7-10 中，并写出其对应的位姿矩阵。

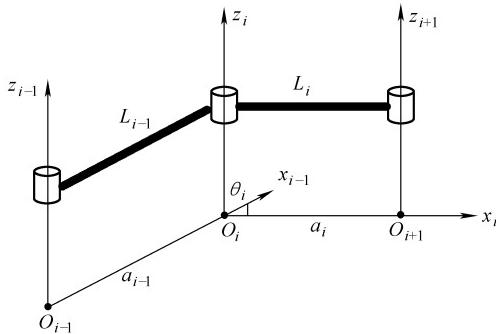


图 7-110 连杆模型 I

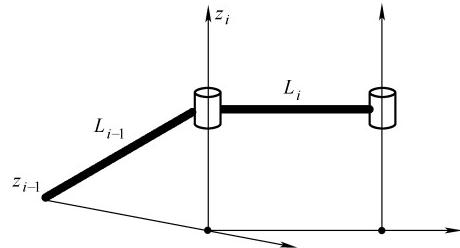


图 7-111 连杆模型 II

注：图 7-111 中 Z_{i-1} 轴与 Z_i 轴是垂直的。

表 7-10 连杆模型 II 参数值

| α_{i-1} | a_{i-1} | d_i | θ_i |
|----------------|-----------|-------|------------|
| | | | |

将其参数值带入公式 (7-7) 中：

$$T_i^{i-1} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ c\alpha_{i-1}s\theta_i & c\alpha_{i-1}c\theta_i & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\alpha_{i-1}s\theta_i & s\alpha_{i-1}c\theta_i & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

带入结果为

$$T_i^{i-1} = \begin{bmatrix} () & () & () & () \\ () & () & () & () \\ () & () & () & () \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2. 机器人连杆坐标系的建立

我们的实验用双足机器人有着 17 自由度，具体自由度配置（对每一个舵机进行了编号，以便

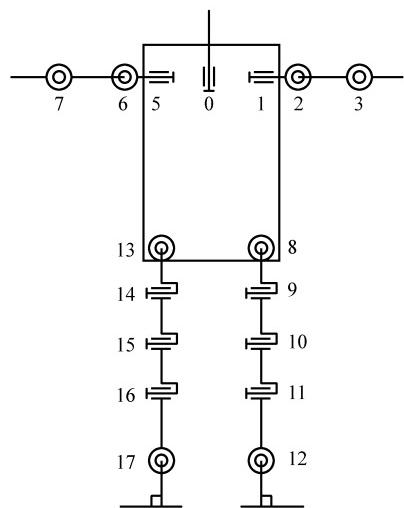


图 7-112 机器人自由度配置图

操作和控制。头部舵机本为 4 号，但在连杆坐标系的建立过程中，我们将头部作为参考坐标系，为便于理解，因此将序号设为 0。如图 7-112 所示。

图中，◎表示舵机在水平面转动；

■表示舵机在垂直面转动。

下面，我们以右臂为例，建立其坐标系，并列出对应的位姿矩阵。

根据图 7-113，将参数填入表 7-11 中。

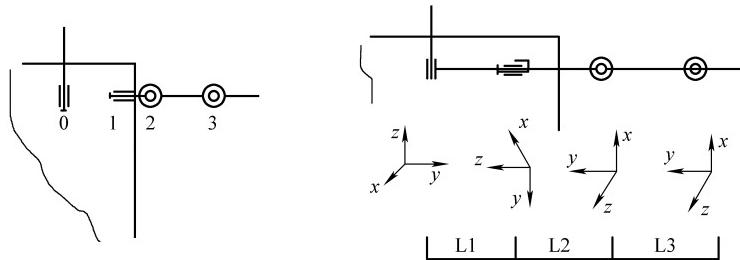


图 7-113 右臂部分坐标系

表 7-11 右臂参数值

| 参数 标号 | α_{i-1} | a_{i-1} | d_i | θ_i |
|----------|----------------|-----------|-------|------------|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |

将表 7-11 中的参数带入式 (7-7)，则

$$T_1^0 = \begin{bmatrix} () & () & () & () \\ () & () & () & () \\ () & () & () & () \\ () & () & () & () \end{bmatrix}, \quad T_2^1 = \begin{bmatrix} () & () & () & () \\ () & () & () & () \\ () & () & () & () \\ () & () & () & () \end{bmatrix},$$

$$T_3^2 = \begin{bmatrix} () & () & () & () \\ () & () & () & () \\ () & () & () & () \\ () & () & () & () \end{bmatrix}。T_2^0 = T_1^0 T_2^1，注：2 号质点相对于 \{0\} 的位姿矩阵。$$

$$T_3^0 = T_1^0 T_2^1 T_3^2 = T_2^0 T_3^2。注：3 号质点相对于 \{0\} 的位姿矩阵。$$

3. 使用 MATLAB 编写机器人模型程序

1) 如图 7-114 所示，启动 MATLAB 应用程序，在 Current Directory 窗口中列出了 MATLAB 默认路径中包含的所有文件，其中有一个 robot1.m 文件。选择此文件双击，或者在 MATLAB 主窗口中选择打开，然后选择 robot1.m 文件打开。

2) 如图 7-115 所示，在 robot1.m 文件中，编写右臂的位姿矩阵部分代码。

3) 打开上位机控制软件，在界面中单击“浏览”按钮，选择一个机器人行为数据，例如，走.dat 文件。选择文件后的界面如图 7-116 所示。

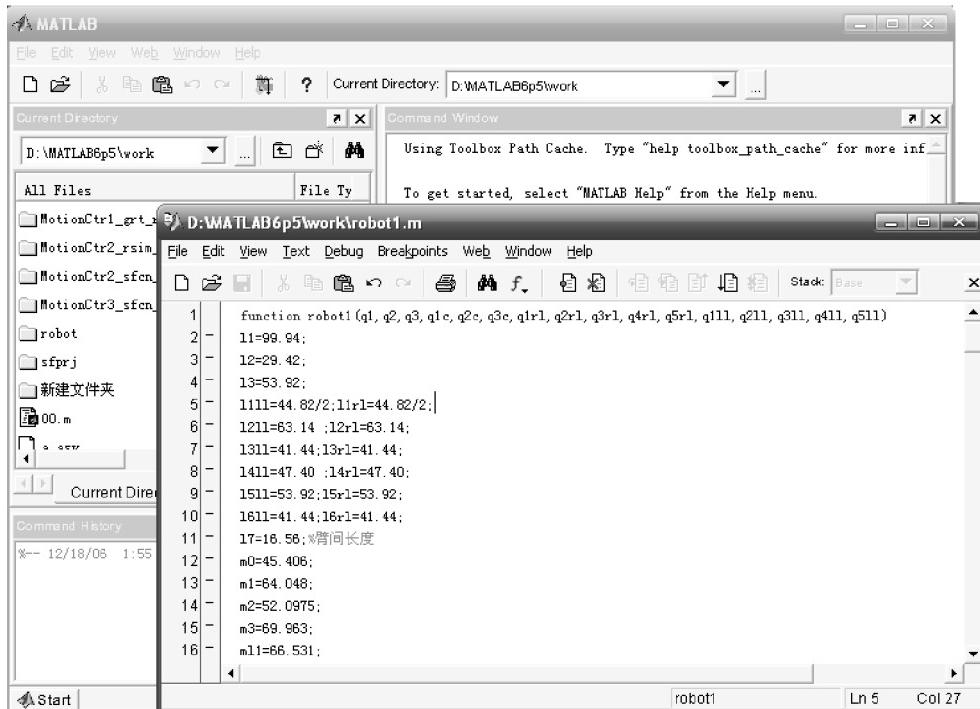


图 7-114 启动并打开 MATLAB 模型文件

The screenshot shows the MATLAB code editor with the file 'D:\MATLAB6p5\work\robot1.m' open. The code defines several variables and parameters. A large oval highlights a section of code starting at line 43, which includes a note in Chinese: '请在下面填写右臂的位姿矩阵' (Please fill in the transformation matrix for the right arm). This section contains code for defining rotation matrices T01, T12, T23, and T03, followed by a call to 'T01c'. The code editor's status bar at the bottom shows the file name 'robot1' and the current line number 'Ln 5'.

```

34 - q3rl=q3rl/180*pi;
35 - q4rl=q4rl/180*pi;
36 - q5rl=q5rl/180*pi;%对舵机的每个角度变量把角度变成弧度
37 - zy=46.25;
38 - zx1=12.80;
39 - zx2=32.46;%脚掌的一些参数
40
41
42
43 %请在下面填写右臂的位姿矩阵
44 %
45 - T01=[  ;  ;  ;  ];
46 - T12=[  ;  ;  ;  ];
47 - T02=  ;
48 - T23=[  ;  ;  ;  ];
49 - T03=  ;
50
51 %
52
53
54
55 - T01c=[0 0 -1 -1;-sin(q1c) -cos(q1c) 0 0;-cos(q1c) sin(q1c) 0 0;0 0 0 1];

```

图 7-115 编写机器人位姿矩阵代码



图 7-116 上位机控制软件界面

这些数据是已有的机器人行为数据，因此每一组数据都应满足稳定性条件，用 MATLAB 仿真，则 ZMP 点都应落在稳定裕内。我们随意选择其中的一组数据，例如第 2 组，则舵机数据为

[13, 19, 97, 179, 168, 98, 97, 85, 72, 115, 95, 100, 99, 116, 74, 92]

从上面的数据可以看到，我们在选择舵机数据时没有包括 CH4 的数据，这是因为 CH4 是头部的舵机，而我们将头部作为了重心模型的坐标原点。

4) 如图 7-117 所示，在 MATLAB 的命令窗口中输入

robot1 (13, 19, 97, 179, 168, 98, 97, 85, 72, 115, 95, 100, 99, 116, 74, 92)
回车后，观察其仿真结果。

5) 输入命令后回车，将会出现 MATLAB 仿真图形，并根据仿真图形完成以下结论。

结论：①输入的角度数据在下载到机器人中时，机器人是否处于稳定状态。即，数据本身是否是可以让机器人稳定的数据。（是或否）

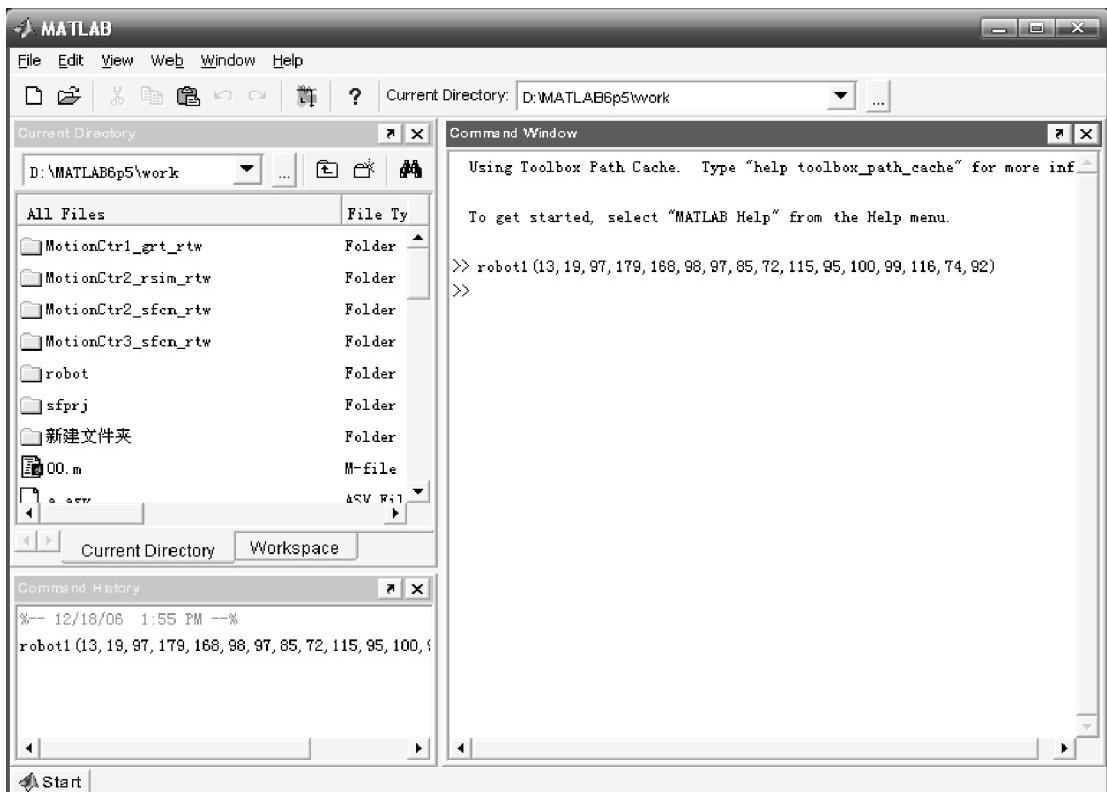


图 7-117 在 MATLAB 主界面输入命令

②仿真图形中，重心在稳定域中的投影是否在稳定域之内。（是或否）

6) 当我们改变行为数据中的一个或多个数据后再输入到 MATLAB 中进行验证，看看改变后的行为数据是否还能满足稳定性条件。例如，我们将上面的数据改为：robot1 (13, 19, 97, 179, 168, 98, 97, 85, 72, 115, 95, 100, 99, 116, 74, 92)，改变了两个数据后再次进行仿真。

结论：①输入的角度数据在下载到机器人中时，机器人是否处于稳定状态。即数据本身是否是可以让机器人稳定的数据。（是或否）

②仿真图形中，重心在稳定域中的投影是否在稳定域之内。（是或否）

五、思考题

1. 选择头部作为坐标原点有什么好处？

答：对于本文中研究的每一个模型，坐标系建立是非常重要的，它的选取直接影响到整个模型的计算量大小、模型的实现、模型的稳定性，所以坐标系的建立在整个模型中是非常基础的，在重心模型中之所以选取头部为坐标系的原点，是基于以下两点原则选取的。

1) 由于重心模型的建立是为了以后更好地实现步行控制，那么在步行控制中，坐标原点应选取在机器人的某个位置，这样可以使计算量尽可能小，否则，如果选取地面上的某个位置作为坐标原点，随着机器人的运动，机器人的每一点的坐标的绝对值将增大，这样模

型的计算量就会大大增加。

2) 在机器人的步行运动过程中,头部一般只是较之前一状态发生平移,不发生旋转,所以选择头部为坐标系原点,计算量就会尽可能减小。

2. 机器人有 17 台舵机,在用 MATLAB 仿真行为数据的稳定性时,为什么输入的舵机角度参数只有 16 个呢?

答:我们在选择舵机数据时没有包括 CH4 的数据,这是因为 CH4 是头部的舵机,而我们将头部作为了重心模型的坐标原点。

3. 建立基础坐标系时, X、Y、Z 轴的方向选取是唯一的吗?选择的不同会有什么影响?

答:建立的坐标系中轴方向的选取并不是唯一的,不过根据基本的规则去选择轴的方向会减少很多不必要的计算量,简化模型。并且结果不会受到影响。

实验十 多智能体协调控制

一、实验目的

- 1) 进一步熟悉上位机软件对类人双足步行机器人的控制。
- 2) 掌握多智能体控制的思想。
- 3) 发挥同学们的创作能力,编排经典的“千手观音”部分动作。

二、实验设备

PC 一台(装有 VC 程序,双足步行机器人上位机控制软件)、无线通信模块/串口通信线、双足步行机器人一组五台、充电电池五套。

三、实验原理与说明

智能体 Agent 是指那些宿主于复杂的动态环境,自主感知环境中信息,自主采取行动,并实现一系列预先设定的目标或任务的计算机系统。它具有推理和学习能力、自主与协同工作能力以及在所处环境中的灵活访问和迁移机制以及同其他智能 Agent 的通信和协调机制,概括起来就是智能性、代理性和机动协调性。该技术最早出现在人工智能领域,其最大特点是具有一定的智能及良好的灵活性和坚定性,特别适合处理复杂、难以预测的问题。基于智能体的控制模型是一种拟人化的模型,可将控制系统的行和其他扩展单元的行为统一起来构成分布式多智能体系统。

多智能体系统 (Multi – Agent System) 是由多个可计算的智能体组成的集合,它能协调一组自主体的行为(知识、目标、方法和规划等),以协同地动作和求解问题。学习是多智能体系统的重要特征之一,多智能体学习可以看做是多智能体系统和机器学习等研究领域的交叉。强化学习是一种以环境反馈作为输入的适应环境的机器学习方法。它不需要对环境的先验知识具有无监督的自适应能力,因而被认为是设计智能 Agent 的核心技术之一。

多智能体系统是针对一群不同种类的、自治的智能体的群体行为所进行的研究,这些智能体可能为同一目标共同工作,也可能为存在潜在冲突的各自不同的目标工作。在具体的研究工作中,多个智能体之间如何协调运用各自的知识、目标、技能和计划共同采取行动、解

决问题是一个重要的研究内容。多机器人系统为多智能体系统理论应用于实际环境中提供了实验平台，同时多智能体系统理论的研究为多机器人系统和多机器人协调协作提供了理论基础。

对于多个机器人的控制，就涉及多智能体（Multi-Agent）控制的思想。每个机器人其实对应的就是其中的单个智能体，我们通过上位机程序和无线通信模块来对多个机器人进行控制，实现了一对多的多智能体控制。

机器人都有自己的 ID 号，每个 ID 号就如同它们自己的名字。在上位机界面中，当我们输入 ID 号时，相应的机器人就会按照所设定的动作组号做出相应动作，这就实现了“一对一”的控制。那对于一对多的控制，我们规定了机器人公用的 ID 号，当我们输入公用 ID 号时，所有开启电源的机器人都会受到控制，这就是所谓“一对多”的控制。本实验分为实验（一）和实验（二），实验（一）体现了一对多智能控制的延时协同控制；实验二体现了一对多智能控制的实时协同控制。

在做这个实验的时候，我们要求同学分组，每组六名，并且配备五台双足类人机器人。动作数据调试完毕之后，将五台双足类人机器人按一列站位，这也就是表演“千手观音”的站位，如图 7-118 所示。

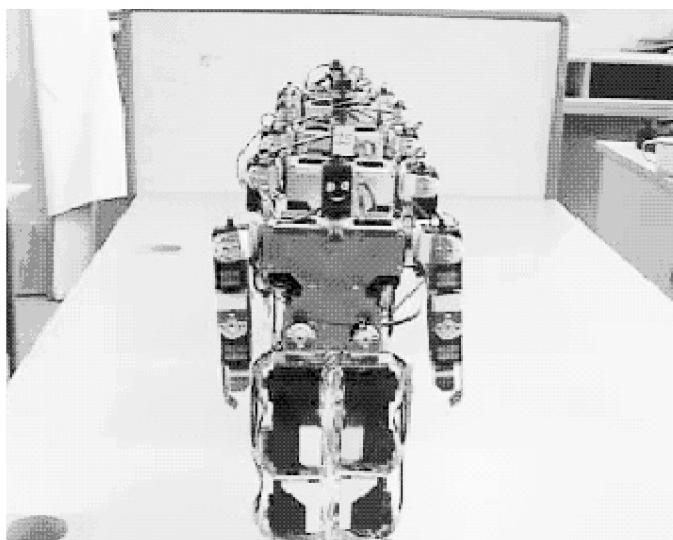


图 7-118 双足机器人表演“千手观音”的站位示意图

四、实验步骤

实验（一）编排千手观音动作之一——“开佛手”

1) 对于千手观音的表演来说，最吸引人的莫过于“开佛手”，如图 7-119、图 7-120 所示的是千手观音中开佛手的动作，具有观赏性，有比较好的演示效果。

2) 关于“开佛手”的动作，我们通过看图片，就可以发现每个机器人的动作其实是一样的，都是将双臂慢慢展开到头上方形成拱形，然后再将双臂放下。那么如何使得每个机器人的动作都会一致呢？

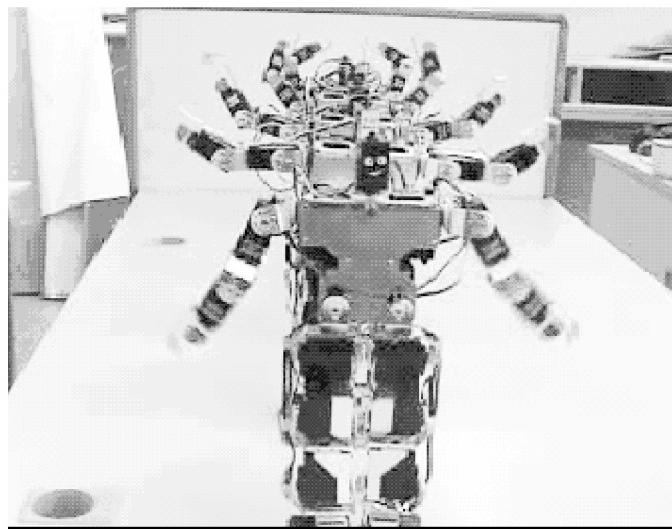


图 7-119 双足机器人表演“千手观音”之“开佛手”示意图一

3) 如果要让 5 个机器人的动作能一致，就只有临摹法（即用一个机器人来做模板，其他机器人照样子学，机器人站立顺序可由动作编排者自行排列）。先将第一个机器人设计展开佛手的动作，这个时候只考虑这个机器人，不要考虑其他机器人。我们可以直接将机器人的“开佛手”动作分为几部分。

动作一：初始位置（Homeposition），它是每个动作最开始的准备部分。具体操作步骤：单击“设定初始位置→读取初始位置数据”，打开初始位置为名的文件夹，读取第一个机器人的初始位置数据（如 1.hos），注意在上位机程序界面中速度控制设为 4，单击“添加”按钮，将初始位置数据放置在数据显示区域中，作为将要调试的开佛手动作的开始。

动作二：拖动 2、3、6、7 号舵机对应的滑块将机器人双臂举到头顶，形成如图 7-120 所示第一个机器人的拱形动作，注意在上位机程序界面中速度控制设为 4，单击“添加”按钮，将动作二的数据放置在数据显示区域中。

动作三：初始位置（Homeposition），它又是每个动作的最末端的结束部分，具体操作步骤如动作一。

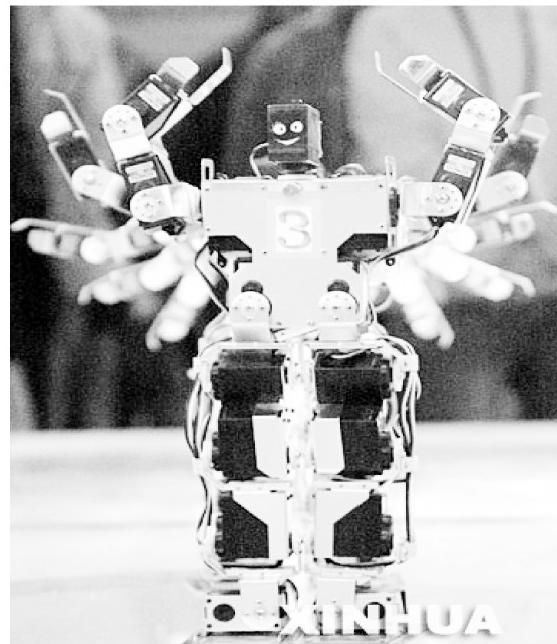


图 7-120 双足机器人表演“千手观音”之“开佛手”示意图二

我们先把第一个机器人的这三个动作实现了。然后单击“保存”按钮，将此数据以“开佛手”为名保存在以机器人 ID 号（如 1 号）为名的文件夹中。

4) 接下来，我们将第一个机器人作为模板，让其他 4 个机器人所编的动作一、动作二都和第一个机器人的动作相同，速度控制也设为 4，重复“步骤 3”。在这个环节，要求 5 个机器人上述“步骤 3”的动作二相同，基本没有偏差。具体作法：读取第一个机器人动作二的数据，让其做动作二，然后将第一个机器人断电使其保持动作二，然后以第一个机器人作为模板来调其他 4 个机器人的动作二。同时我们将上位机程序里的 ID 号设为公共 ID 号 127，使得 5 个机器人一起做动作二。

5) 最后是要把延时加入动作中，因为“开佛手”需要每个机器人的动作有一定的延时性，这样才保证具有图 7-120 的效果。首先对于第一个机器人，要将动作二的动作数据持续八行，使得动作二能持续一段时间。而动作一、动作三都为初始位置，不需要加延时，只要一行就够了。具体操作步骤：单击“浏览”按钮，在以机器人 ID 号（如 1 号）为名的文件夹中读取第一个机器人“开佛手”的动作数据，双击动作二对应的动作数据行，然后再单击动作二对应的动作数据行，再单击“插入”按钮，然后再单击刚刚插入的动作数据行，再单击“插入”按钮，如此进行下去插入 7 行，将动作二的动作数据持续 8 行。延时加完之后，将数据保存后，将该组开佛手的动作数据通过无线或是有线的方式发送给第一个机器人，单击上位机控制界面下端的“发送”按钮〔在单击发送之前要对控制界面中的“组号”进行配置，组号可以自行设定，如“开佛手”的动作组号设为 6，ID 号为第一个机器人的 ID 号（如 1 号）〕，将已调试好的开佛手动作发送至下位机，出现对话框后单击“确定”，之后单击“运行”按钮（同样在单击之前也要选择相应的组号），第一个机器人将完成开佛手动作。

6) 对于第二个机器人，它是紧跟第一个机器人做动作的，这时需要在动作一中有持续，来实现机器人动作的延时。于是，将第二个机器人的动作一的动作数据持续 15 行，达到延时效果，具体操作步骤：单击“浏览”按钮，在以机器人 ID 号（如 2 号）为名的文件夹中读取第 2 个机器人“开佛手”的动作数据，双击动作一对应的动作数据行，然后单击第一行数据，再单击“插入”按钮，然后再单击刚刚插入的动作数据行，再单击“插入”按钮，如此进行下去插入 14 行，将动作一的动作数据持续 15 行。然后将动作二还是持续 8 行，即双击动作二对应的动作数据，然后再单击动作二对应的动作数据行，再单击“插入”按钮，然后再单击刚刚插入的动作数据行，再单击“插入”按钮，如此进行下去插入 7 行，将动作二的动作数据持续 8 行。动作三还是保持一行不变。延时加完之后，将数据保存后，将该组开佛手的动作数据通过无线或是有线的方式发送给第 2 个机器人，发送时组号可以自行设定，但必须为 0~20 之间未用过的数字（如“开佛手”的动作组号设为 6），ID 号为第 2 个机器人的 ID 号（如 2 号）。

7) 对于后面的第 3 个、第 4 个、第 5 个机器人，同样的需要将动作一分别持续 25 行、35 行、45 行，分别实现延时的效果。而对于这后面的 3 个机器人，动作二都需要持续 8 行，动作三都还保持一行不变，具体操作步骤如步骤 6)。

8) 最后将 5 个机器人摆放成前面所调的机器人的顺序站成一行，并且打开电源，即如图 7-118 所示的“千手观音”的站位队列，程序界面的 ID 号设置成公共 ID 号 127，然后运行，我们可以看到“千手观音”之“开佛手”动作就会表演出来。

9) 当然如果效果不是最佳, 可以考虑在某个机器人要持续的动作的行数上改动几行(增加或减少几行), 因为每个机器人都会存在差异, 效果如图 7-120 所示为最佳。

实验 (二) 编排千手观音动作之二——“展臂”

1) 对于千手观音的表演来说, “展臂”也是很重要的一环, 要求所有机器人同时做动作, 而且每个机器人向不同方向展臂, 如图 7-121、图 7-122 所示。

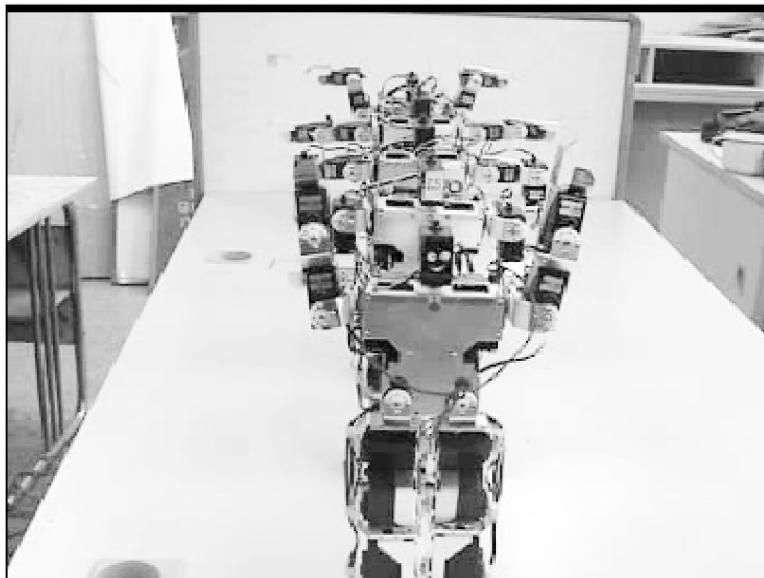


图 7-121 双足机器人表演“千手观音”之“展臂”示意图一

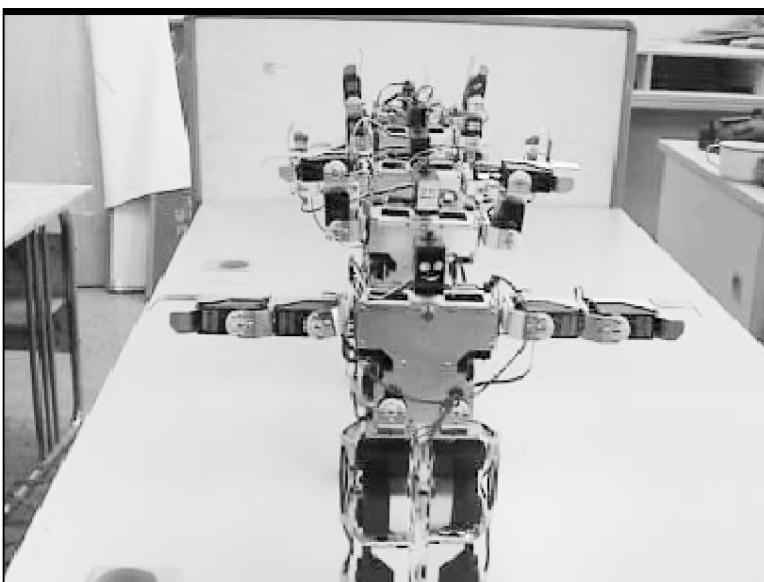


图 7-122 双足机器人表演“千手观音”之“展臂”示意图二

2) 关于“展臂”的动作，我们通过看图片，就可以发现每个机器人的动作是不一样的，都是瞬间从初始位置转到一定方向，让手臂保持某个特定动作一段时间，然后再将双臂转回初始位置。

3) 我们可以直接将机器人的“展臂”动作分为4部分。

动作一：初始位置，它是每个动作的最开始的准备部分，如对第一个机器人具体操作步骤如“实验（一）”中步骤3)的动作一。

动作二：如图7-123所示每个机器人对应的动作。如对于第一个机器人，拖动2、3、6、7号舵机对应的滑块将机器人双臂竖直举过头顶，形成如图7-123所示第一行机器人的动作，单击“添加”按钮，将动作二数据放置在数据显示区域中。

动作三：如图7-124所示每个机器人对应的动作。如对于第一个机器人，拖动2、3、6、7号舵机对应的滑块将机器人双臂举平，形成如图7-124所示第一行机器人的动作，单击“添加”按钮，将动作三的数据放置在数据显示区域中。

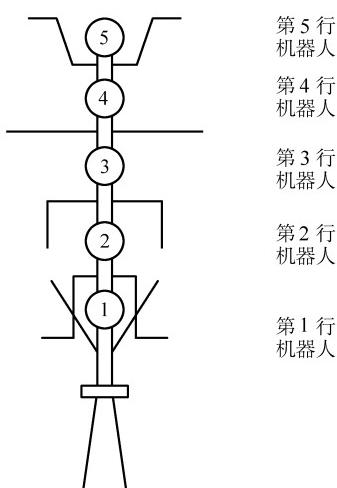


图7-123 5个机器人动作二的正面示意图

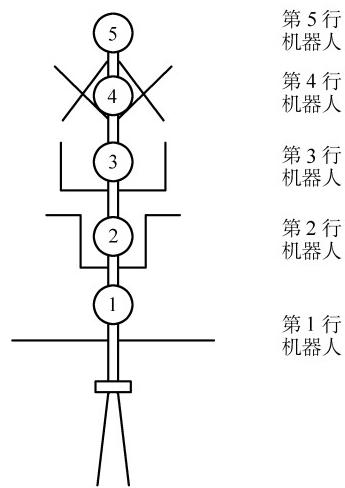


图7-124 5个机器人动作三的正面示意图

动作四：初始位置，它又是每个动作的最末端的结束部分，如对于第一个机器人具体操作步骤如“实验（一）”中步骤3)的动作三。

然后单击“保存”按钮，将此数据以展臂为名保存在以机器人ID号（如1号）为名的文件夹中。同理，其他四个机器人也如上述4个部分来编排展臂动作，不同的是他们的动作二、动作三应分别对应图7-123和图7-124所示的每个机器人的动作来调。

4) 由于“展臂”需要5个机器人同时做动作，所以这里规定每个机器人都要遵守下列规定：动作一要保持的行数为4行（速度控制设为4），动作二要保持的行数为40行（第一行速度控制设为7，其余都设为1），动作三要保持的行数为40行（第一行速度控制设为7，其余都设为1）、动作四要保持的行数为1行（速度控制设为7），按照上述规编定写动作，将体现一对多智能控制的实时控制。

例如对于第一个机器人，要将动作一的动作数据持续4行，具体操作步骤：单击“浏

览”按钮，在以机器人 ID 号（如 1 号）为名的文件夹中读取第一个机器人“展臂”的动作数据，双击动作一对应的数据行，速度控制设为 4，然后再单击动作一对应的数据行，再单击“插入”按钮，然后再单击刚刚插入的数据行，再单击“插入”按钮，如此进行下去插入 3 行，将动作一的动作数据持续 4 行。要将动作二的动作数据持续 40 行，双击动作二对应的数据行，速度控制设为 7，然后再单击动作二对应的数据行，单击“修改”，然后再速度控制设为 1，单击动作二对应的数据行（速度控制已设为 7）的下一行，再单击“插入”按钮，然后再单击刚刚插入的数据行，再单击“插入”按钮，如此进行下去插入 39 行，将动作二的动作数据持续 40 行，同理为动作三按上述规则加延时。最后将动作四的数据速度控制修改为 7。延时加完之后，将数据保存后，将该组展臂的动作数据通过无线或是有线的方式发送给第一个机器人，发送时组号可以自行设定，但必须为 0 ~ 20 之间未用过的数字（如“展臂”的动作组号设为 7），ID 号为第一个机器人的 ID 号（如 1 号）。其他机器人同理。

5) 关键部分在于每个机器人动作二、动作三的编写。每个机器人的动作二、动作三都不同，且要求在编写动作二时，五个机器人虽然动作不相同，但总体上看，如图 7-121 所示，手臂的空间分布上下均匀。动作三同理。

6) 在本实验中，特定的动作二、动作三不拘泥于一种形式，也就是同学们编写的时候，可以不参照图 7-121 与图 7-122，自己编写，但要总体上遵循手臂的空间分布上下均匀，具有层次感，这样效果才会更好。在此注意一点，对于同一个机器人，动作二、动作三应该对应：比如动作二机器人手臂上抬，动作三机器人手臂就下垂。

7) 当每个机器人都编排好动作后，将 5 个机器人摆放成前面所调的机器人的顺序站成一行，并且打开电源，即如图 7-118 所示的“千手观音”的站位队列，然后程序界面的 ID 号设置成公共 ID 号 127，接着运行，我们可以看到“千手观音”之“展臂”动作就会表演出来。

8) 当然如果效果不是最佳，可以考虑在某个机器人要持续的行数上改动几行（增加或减少几行），因为每个机器人都会存在差异。

五、思考题

1. 机器人的“千手观音”动作表演中的延时是怎样实现的？

答：在机器人“千手观音”动作表演中的延时是通过对上位机界面中的某行数据进行复制，达到动作持续的效果。在这里，我们一般会用到两种按钮：“插入”、“添加”。

其中“插入”按钮用得最普遍，当我们要复制某行数据时，首先先双击这行数据，这时发现滚动条也随之显示该行数据，然后单击要插入的位置的下一行，再单击“插入”按钮，想要复制的行就复制了一行，由此想要复制几行，就单击“插入”按钮几下。实现动作的延时，当然复制的行数越多，延时越长。

“添加”按钮用得很少，一般都是在要插入的位置后面已经没有任何数据的前提下，单击“添加”按钮，想要复制的行就在数据的最后一行复制了一行。由此想要复制几行，就单击“添加”按钮几下，注意的是每次添加的行都是在最后一行，以前复制的行都向上推进。

2. 我们是如何用一个上位机程序来同时控制多个机器人的？

答：对于一个上位机程序对多个机器人的控制，我们规定了机器人公用的 ID 号，当我

们输入公用 ID 号时，所有开启电源的机器人都会受到控制，这就实现了一个上位机程序对多个机器人的控制。

3. 我们从机器人的“千手观音”动作中的“开佛手”和“展臂”可以得出什么结果，利用这个结果，我们如何编排新动作？

答：对于机器人的“千手观音”动作中的“开佛手”，我们可以得到 5 个机器人依次延时地做动作时，动作变化的具体行数，也就是说，当我们设计新的动作，而且这个动作要求机器人是依次延时地去做，我们可以直接利用“开佛手”得到的结果，知道每个机器人在具体哪一行要换动作的前提下，将动作数据编写上去就可以了。

同理，对于机器人的“千手观音”动作中的“展臂”，我们可以得到五个机器人同时做动作时，动作变化的具体行数，也就是说，当我们设计新的动作，而且这个动作要求机器人是同时地去做，我们可以直接利用“展臂”得到的结果，知道每个机器人在具体哪一行要换动作的前提下，将动作数据编写上去就可以了。

这样，我们可以利用“开佛手”和“展臂”的结果，编排任何动作，包括延时地和同时地做动作。

4. 机器人团队除了“千手观音”动作之外，还适合做什么样的团队表演，发挥想象列举一些其他的团队表演。

答：机器人团队除了“千手观音”动作之外，还可以表演一些团队表演，例如团队街舞表演、团队体操表演、团队扭秧歌表演、团队抬花轿表演、团队武术表演、团队打太极表演、团队踢球表演、团队竞赛类（跑步、翻跟头、俯卧撑）表演、团队拔河表演等。

实验十一 复杂动作实验

一、实验目的

- 1) 进一步熟悉双足步行机器人上位机控制平台的操作。
- 2) 学习用上位机软件，完成复杂动作——爬楼梯的调试，学会机器人复杂动作的调试方法。
- 3) 掌握发送多组数据的方法。
- 4) 掌握动作编排的方法。

二、实验设备

PC 一台（装有 VC 和双足步行机器人上位机控制软件）、无线通信模块/串口通信线、双足步行机器人一台，电池、充电设备各一套。

三、实验原理与说明

1. 复杂动作的调试方法

在简单动作实验里面，我们已经学会了利用上位机来完成招手动作的调试，接下来我们来介绍一组复杂动作的调试过程。

所谓复杂动作，就是指整个动作全部完成，需要很多帧数据，这些帧顺序执行，可以形

成一组比较完整并稍微复杂些的动作。一般，复杂动作中除机器人上肢运动以外，通常会涉及跨部及下肢各舵机的运动，这样就存在机器人重心偏移后各舵机如何相互配合仍使机器人在完成每一帧数据后保持所需平衡的问题。这里需要注意的是在复杂动作中，机器人完成每一帧动作后都需处于一个稳定的状态（即能维持平衡），并且帧与帧之间的动作差别不应过大，而且在速度位的设置上也需配合动作需要，过快可能造成机器人由于惯性失去平衡，过慢则导致动作的观赏性下降。

2. 让机器人执行多组动作的方法

多动作实际上是由多组简单和复杂动作按照我们事先设计的规则顺序执行的过程。

首先，我们需要把已经调试好的各个动作，顺序地发送到机器人下位机存储器的不同区域中，然后通过上位机软件自带的动作编排窗口进行所需顺序的动作编排，之后单击“顺序”按钮，就可以执行我们所需要的动作序列了。具体的设置方法参见下面的实验步骤。

四、实验步骤

1. 调试一组复杂动作——爬楼梯（这里我们只以上一级台阶为例进行讲解，上多级台阶方法可由一级台阶调试方法进行推广）

1) 打开机器人电源开关，上电后机器人处在初始位置（双脚直立状态），先将机器人放在自制的楼梯前，做好下一步调试准备，并单击上位机界面上的“设定初始位置”→“读取初始位置”调出已调试好的机器人初始位置数据，并单击“添加”按钮，把初始位置数据作为机器人爬楼梯动作的第一帧，如图 7-125 所示。

2) 将头脑中人类爬楼梯的动作分解成一个个更细小的动作，也就是我们这里指的“帧”。首先假设我们人直立站在楼梯台阶前，下一步动作应该是抬脚，即一只脚抬起来，另一只脚站立，且要保持平衡。双足机器人也是一样，下一帧数据就是要让机器人一脚站立，另一只脚离开地面，这中间不可能只有离开地面的那个舵机在动，因为要平衡，另一只脚跨部舵机，脚腕部舵机等都需要调整。调整完这一帧数据后，机器人仍需保持平衡不倒。

这里，我们首先令右脚作为机器人支撑脚，左脚抬起，作为爬楼梯动作第 2 帧。

要完成这个动作，我们分别通过滑块修改左腿跨部舵机 CH8、右脚跨部舵机 CH13 参数，修改范围大约为 95~110 之间，使机器人重心转移到右腿上。然后修改左腿腿部各舵机 CH9、CH10、CH11、CH12 和右腿腿部各舵机 CH14、CH15、CH16、CH17 参数，使机器人左腿抬起且右腿能够作为支撑腿保证机器人单腿站立而不倒。调试完成后，单击“添加”按钮，将这一帧添加到数据显示区域中，如图 7-126 所示。

3) 修改左腿腿部各舵机 CH9、CH10、CH11、CH12 参数，使机器人完成“右腿不动，左脚放在台阶上”的动作，作为第 3 帧数据添加到数据显示区域，方法同上。

4) 修改左腿跨部舵机 CH8、右脚跨部舵机 CH13 参数，修改范围大约为 70~85 之间，使机器人重心转移到左腿，并修改左腿腿部各舵机 CH9、CH10、CH11、CH12 和右腿腿部各舵机 CH14、CH15、CH16、CH17 参数，令机器人完成“重心由台阶下的右脚转移到台阶上的左脚”的动作，作为第 4 帧数据添加到数据显示区域。

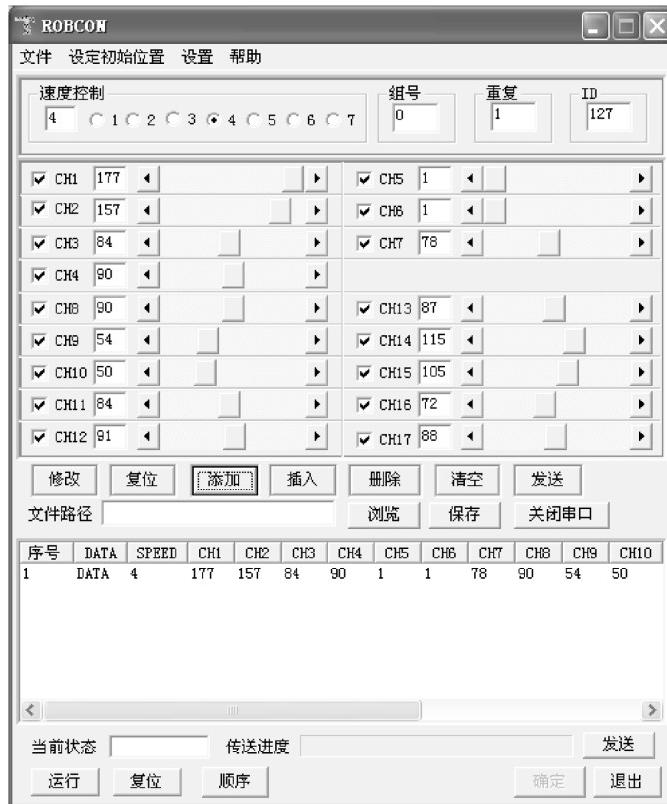


图 7-125 设置机器人初始位置为爬楼梯动作第一帧



图 7-126 添加抬腿动作作为爬楼梯动作的第二帧

5) 修改右腿腿部各舵机 CH14、CH15、CH16、CH17 参数，使机器人完成“左脚不动，台阶下的右脚抬起”的动作使右脚离开地面，整个机器人由台阶上的左腿单独支撑。将调试好的数据作为第 5 帧数据添加到数据显示区域。

6) 修改右腿腿部各舵机 CH14、CH15、CH16、CH17 参数，使机器人完成“将已抬起的右脚放到台阶上”的动作，此时机器人重心仍在左腿上，右脚轻放在台阶上，不作支撑机器人重量使用。将调试好的数据作为第 6 帧数据添加到数据显示区域。

7) 将机器人的初始位置（双脚直立状态）数据，作为上台阶动作的第 7 帧添加到数据显示区域，方法同第一帧数据的添加方法。

调试一个复杂动作的过程就是上面的一个过程。

8) 我们在调试完整个分解过程后，就需要把这些分解后的动作连起来，也就是我们把整个一组数据发到单片机里后，看看效果如何。

首先单击“发送”按钮，将动作数据下载到机器人处，然后单击“复位”按钮让机器人先复位到初始位置，最后单击“运行”，察看所调试的完整动作。

有时你会发现，单个动作都没有问题，但连起来会出问题，也就是衔接不好。这可能是两方面的原因：

① 动作分解得不够细，中间动作丢失，失去了衔接；

② 动作速度位不对，比如由一组动作切到另一组，由于速度太快，机器人有惯性，也会失去平衡。

在这种情况下，继续细调数据，使之更好地衔接并使速度设置更合理些即可。

同样的，有时由于动作分解过细，也会出现很多冗余帧，使数据量变大。实际上没有这些数据，机器人同样可以很好地完成动作，为了简单化，应删掉多余的数据帧。

调试过程中，有时不知道该给每个舵机调多少才能到达某种状态并保持平衡，我们可以采取先断电，用手把机器人扳倒那种状态，大概观察下舵机的位置，然后再通电，利用上位机同步模式来拉动滑块，使舵机到达记录的那种状态，之后再进行细调。

9) 调试好的动作，单击“保存”按钮，弹出如图 7-127 所示对话框，将文件名设定为“上台阶.dat”，然后单击右边的“保存”按钮。

2. 让机器人执行多组动作

(1) 多组动作发送 多动作也是复杂动作的一种，比如我们想让机器人先走路，再爬楼梯，再翻跟头等等，这就需要我们事先调试好这些单独的动作，再依次发到单片机里，我们注意到上位机程序右上角有个组号，这个组号就是表示数据在下位机中存放的区域，如图 7-128 圈中标注所示。

1) 先向单片机里发送第一组数据：走路。方法如下：单击“浏览”，选择已经保存好的走路数据，之后单击“打开”按钮，如图 7-129 所示。

2) 然后单击“发送”按钮，将走路数据发送到下位机第 0 号数据存储区域中。

3) 同上述步骤，再向下位机第 1 号动作存储区域中发送爬楼梯数据。唯一不同的是在单击“发送”按钮之前，要把组号框中的“0”改成“1”，之后再单击“发送”以后各动作以此类推，均发送到不同的指定区域中。如果不改组号，则存储区域中的数据将被新发送过来的数据所覆盖，以最后一次发送的动作为准。



图 7-127 保存已调试好的机器人动作数据



图 7-128 上位机软件组号标注

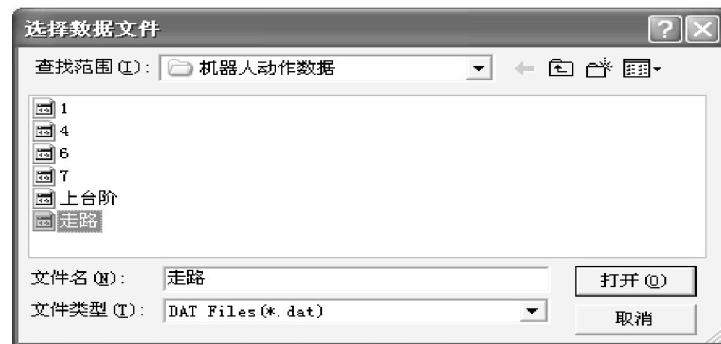


图 7-129 打开走路数据文件

(2) 动作组编排 将已经存储到机器人下位机中的各组动作，通过动作组编排方式，以我们所需要的顺序，控制机器人来执行。比如我们存放时顺序为：1、走路；2、爬楼梯；3、翻跟头；4、踢球；5、鲤鱼打挺；6、招手；7、摇头；8、俯卧撑（注意：走路为第1套动作，其存放在第0组，即组号为0。依此类推，第8套的俯卧撑存放在第7组中，组号为7）。现在让它执行6、招手→1、走路→7、摇头→4、踢球。方法如下：

1) 单击菜单中的“设置”→“动作顺序”，弹出如图7-130所示窗口。

2) 依次双击左侧框中的“动作6”、“动作1”、“动作7”、“动作4”即可，注意左边框里数据的变化，右下角的重复框中，填入“1”如图7-131所示。

3) 编排窗口右面1~4区域中对应的数据为6、1、7、4，然后单击“确定”。回到主界面，再单击上位机软件下部的“顺序”按钮，机器人即可按要求依次执行6、1、7、4四个动作。

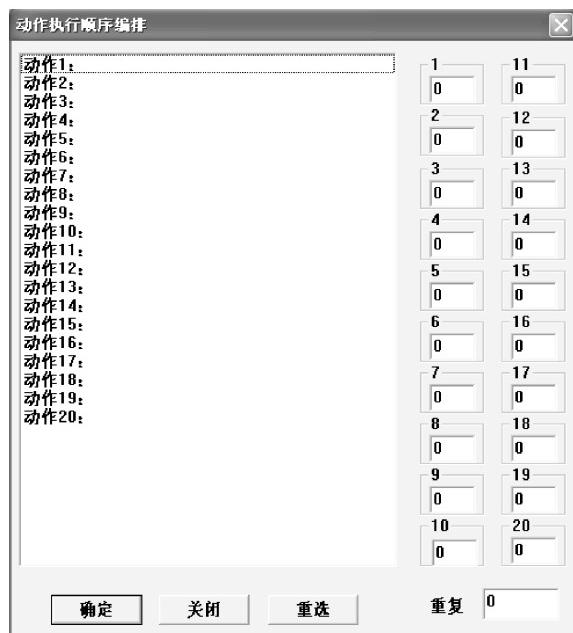


图 7-130 动作执行顺序编排窗口未进行设置

五、思考题

1. 调试出一个复杂动作的方法有哪些？

答：(1) 通过已经有的机器人录像进行动作分解，单个调试分解动作，然后再在各分解动作中间插入必要的衔接动作。

(2) 不加电自己将机器人摆成所调试动作中关键的几帧，目测每一帧中各个舵机位置，然后加电修改舵机角度，到达这些位置，最后再在中间插入必要的衔接动作。

2. 如何发送多组数据到单片机？

答：修改上位机软件中的数据组号，然后一组组发送即可。

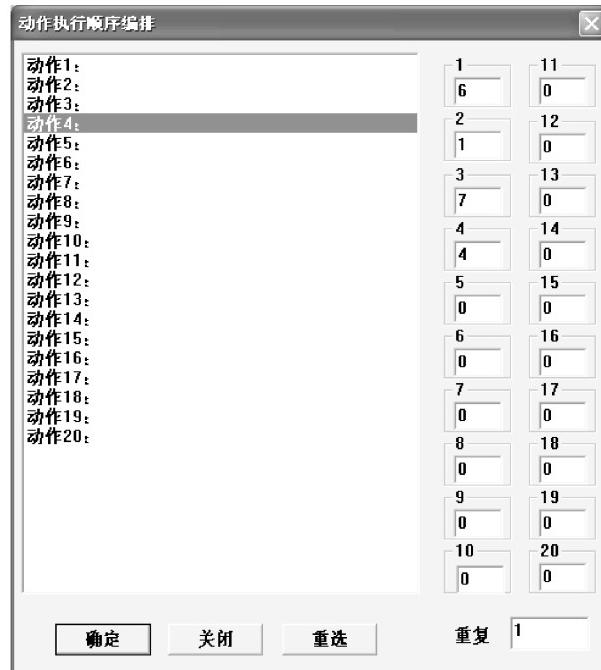


图 7-131 动作执行顺序编排窗口设置完成

3. 怎样让机器人按设定的动作顺序运行动作?

答：通过动作顺序调试设置运行“顺序”即可。

实验十二 语音控制程序实验

一、实验目的

- 1) 掌握双足步行机器人上位机语音控制平台的安装。
- 2) 学会语音控制平台的设置和调试。
- 3) 应用机器人语音控制平台控制机器人完成动作。
- 4) 编写语音识别程序，并调试。

二、实验设备

硬件设备：

PC一台（要求：内存：256MB及以上、CPU：1.8GHz、声卡：独立多功能声卡、麦克：质量中上等的麦克）、串口通信线、双足步行机器人一台、充电电池一块、语音识别加密狗一个。

软件设备：

Windows2000/NT/XP或以上版本操作系统、语音识别SDK（语音识别库——Mandarin_8kHz3.0、asrapi.dll、ASRAPI.h、asrapi.lib）、语音识别加密狗驱动程序、语音合成服务器系统InterPhonic CN Rev1.0版。

三、实验原理与说明

1. 语音识别原理

语音识别分为预处理、特征提取、模式匹配三部分，如图 7-132 所示。

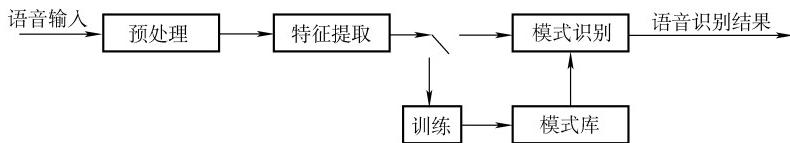


图 7-132 语音识别算法原理框图

不同的语音对应着不同的波形，预处理过程把声音转换成相应的波形，特征提取模块提取波形的波长，频率等特征，如“招手”和“下蹲”这两句话预处理完之后对应着完全不同的波形，根据提取的特征和模型库中的模型进行匹配，得到相应的文字，这样就完成了整个语音识别的过程。

2. 双足机器人语音识别功能模块

双足机器人语音识别功能模块利用语音识别产品 Pattek ASR SDK 在 Visual C ++ 6.0 环境下进行开发。

模式语音 ASR 是国内首屈一指的整套语音识别解决方案，具备中文、英文等多种语言的连续语音识别以及听写功能，适合于开发计算机、电话、嵌入式终端等各种设备上的语音应用程序。

依托中国科学院自动化研究所在语音识别研究上二十年的技术积累，Pattek ASR 具有易用性强，识别率高的特点。用户无须进行训练，引擎的设计已经保证了非特定人这一重要特点；API 提供的管理工具可以使得用户可以自如地定义自己所需的词表和语法，以便应付不同的应用需求。API 就是应用程序编程接口。它是能用来操作组件、应用程序或者操作系统的一组函数。典型的情况下，API 由一个或多个提供某种特殊功能的 DLL 组成。（Java 中有所不同，但大同小异）。DLL 是一个文件，其中包含了在 Microsoft Windows 下运行的任何应用程序都可调用的函数。运行时，DLL 中的函数动态地链接到调用它的应用程序中。无论有多少应用程序调用 DLL 中的某个函数，在磁盘上只有一个文件包含该函数，且只在它调入内存时才创建该 DLL。

双足机器人语音识别功能模块的程序流程图如图 7-133 所示。

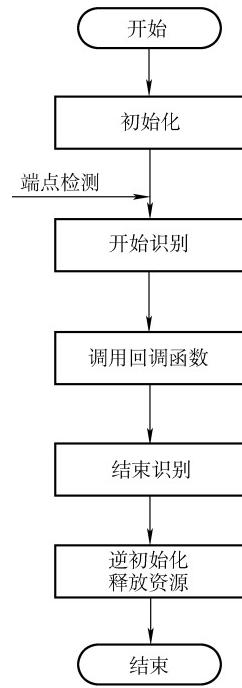


图 7-133 语音识别模块流程图

3. 双足机器人上位机语音识别程序框架

在程序的开始，首先要利用 SDK 提供的 Asr_SetSystemDir（）函数设置识别引擎工作的系统目录，主要是指定声学模型所在的目录，然后利用 Asr_SysInit（）初始化语音识别系统。

如果语音识别系统初始化成功，则可以利用 Asr_LineInit（ResultProc）函数来创建一个识别引擎 hLine，并指定该引擎将结果识别出来时的回调函数以及用户定义数据。

在本文中我们对语音识别回调函数定义为

```
DWORD ResultProc( HASRLINE hLine, const PASR_NOTIFY pNotify, DWORD dwUserData);
```

其中 pNotify 是回调通知，它的结构体如下：

```
typedef struct _ ASR_NOTIFY
{
    int NotifyType;
    int data;
    DWORD dwReserve;
}
ASR_NOTIFY, *PASR_NOTIFY;
```

这个结构在回调函数中返回与语音以及语音识别结果相关的数据，其中 NotifyType 为通知类型，包括：

| | | |
|--------------------|---|---------|
| NOTIFY_RESULT | 1 | //识别结果 |
| NOTIFY_VOICE_LEVEL | 2 | //声音大小 |
| NOTIFY_UTT_BEGIN | 3 | //一句话开始 |
| NOTIFY_UTT_END | 4 | //一句话结束 |

data：当 NotifyType 为 NOTIFY_VOICE_LEVEL 时，data 表示当前录音的大小，NotifyType 为其他值时，data 无意义。

dwReserve：保留，无意义。

然后我们创建录音回调函数，录音回调函数用来记录语音数据形成数据文件，供语音识别回调函数调用。录音回调函数如下：

```
int cb( const char * pbuf, int size)
{
    Asr_RecFillWaveBuffer(hLine, pbuf, size);
    return 0;
}
```

成功创建一个识别引擎后，利用 Asr_GrammarLoad（）函数来导入语法规则文件中的语法规则，并对语法规则进行设置。语法规则文件可以由用户按照语法规则的要求根据需要自己编写，语音识别所需要用到的语法规则，遵循 ABNF 范式（Augmented BNF），兼容 W3C 语法定义。语法规则的结构体如下所示：

```
typedef struct _ GRAMMAR_ACTIVATE
{
```

```

int nNum;
char * szRuleName[ 100 ];
{
HASR _ GRAMMAR, * PHASR _ GRAMMAR;

```

我们可以使用该结构来设置、切换、关闭识别引擎对应的语法规则。其中：nNum 是语法规则的数量；szRuleName 是语法规则的名字，一个语音识别引擎可以设置多个语法规则。

成功导入语法规则后，我们利用 Asr _ GrammarActivate () 函数来激活语法规则，设置 Asr _ RecSetAutoWork (true) 函数由引擎内部自动采集语音数据，然后调用 Asr _ SetParam () 函数设置录音以及引擎相关的参数，最后调用函数 Asr _ RecStartX (hLine) 开始自动识别。

端点检测到语音大于系统设置的起始话音音量并进行录音，当语音小于设置的结束音量停止录音。语音识别系统对语音进行识别，通过回调函数得到回调通知，利用 Asr _ GetNBestResult () 函数获得识别结果。

如果不再需要语音识别，利用 Asr _ RecStopX (hLine) 函数停止由 Asr _ RecStartX 函数开启的识别任务，利用 Asr _ LineRelease (hLine) 函数关闭 hLine 语音识别引擎，并且释放其相关的资源，利用 Asr _ SysRelease () 函数退出系统，清除识别系统申请的资源。

四、实验步骤

实验（一）安装语音控制平台

- 1) 安装语音合成软件 InterPhonic CN Rev1.0 版，安装 iFlyTTS _ SDK，安装加密狗配置程序（在安装盘中）。

- 2) 在系统 DSN 里注册数据源，选择驱动 Microsoft access driver (*.mdb) 建立数据库 ShuangZudb. mdb。具体操作步骤如下：

打开控制面板→管理工具→数据源（ODBC）→系统 DSN→添加→选择 Microsoft access driver (*.mdb)，单击“完成”后→在数据源名里输入 dashendb，同时单击数据库“选择”按钮，在目录项中选择相应路径（如 D: \ \ 语音控制实验 \ Data）下的 ShuangZudb. mdb，并在数据库名的编辑框中填入 ShuangZudb. mdb 即可。

- 3) 把串口 1 和无线发射器连接起来。

- 4) 插上传声器（即麦克风）。

实验（二）熟悉语音控制平台

1. 界面设置

打开文件“语音控制实验”→“语音控制平台”→RobotSoft. exe，就会出现如图 7-134 的主界面。

菜单栏主要包括登录选项、功能设置、界面设置、版本信息、退出（退出密码：000）。

状态栏主要包括成功语音识别初始化信息、语音合成初始化信息、视频初始化信息。

当用户对着话筒按照界面左侧移动的字幕讲话时，如：一号跳街舞，通过语音识别就可以命令机器人做各种动作。

注意：当讲完话后机器人没有任何反应时，可能有两个原因：①串口不一致。解决方法：若有两个串口，必须连接至串口 1 上，若因特殊原因连接至串口 2 上，打开“语音控制



图 7-134 主界面

平台”文件夹，打开文件“ComSet.ini”，将“nPort = 1”改成“nPort = 2”，之后保存即可。

②必须用音效较好的传声器，且不能将音量控制中的传声器的音量调制得太高。一般还要在传声器的“高级”设置里将“Mic Boost”复选上，有少数情况下（如计算机差异的原因）不能复选。

2. 识别语音设置

打开菜单项“登录选项”，单击“登录”，在弹出的对话框中输入登录密码（000），单击“登录”即可看到图 7-135。

系统默认得起始音量是 100，最高音量是 300。用户可以根据需要自由设置音量大小（起始音量 < 最高音量），只要用户说话得音量在这两个数值之间，系统都能成功识别。

实验（三）编程实现语音识别模块

1) 编写语音识别模块 VC 控制平台程序。

打开 Microsoft Visual C++ 6.0，新建一个工程：选 Win32 Console Application，工程名可写为“Yysb”，Location 可以根据需要自己设定，单击 OK，选择“A ‘Hello, World!’ application.”，然后单击“Finish”，单击“OK”。单击工作区中的 Class view 选项卡中的 Yysb classes 前的“+”，再单击 Globals 前的“+”，然后双击 Globals 中的成员函数。然后在右边的代码编辑器中，将注释号“//”下面的内容删除掉，然后编写语音识别模块 VC 控制平台程序。



图 7-135 语音识别音量设置

语音识别模块 VC 控制平台程序示例：

```
#include "stdafx.h"
#include "stdio.h"
#include "iostream.h"
#include "asrapi.h"
#include "windows.h"
#define ONLINE_TEST
HASRLINE hLine;
//录音回调函数
int cb( const char * pbuf, int size)
{
    Asr_RecFillWaveBuffer(hLine,pbuf,size);
    return 0;
}
//语音识别回调函数
DWORD ResultProc ( HASRLINE hLine , const PASR_NOTIFY pNotify , DWORD dwUserDa-
ta)
{
    switch( pNotify ->NotifyType)
    {
        case NOTIFY_UTT_BEGIN:
            break;
        case NOTIFY_UTT_END:
            break;
        case NOTIFY_VOICE_LEVEL:
            break;
        case NOTIFY_RESULT:
            int count;
            char ** result;
            int * confidence;
            Asr_GetNBestResult ( hLine ,count , result ,confidence );
            if( count == 0 )
                printf("拒识\n");
            else
            {
                for( int i =0; i < 1; i++ )
                {
                    printf("\r%dth: %s\n", i +1 , result[ i ] );
                }
            }
    }
}
```

```

    }

#ifndef ONLINE _ TEST
    printf( "请继续说话推出请按' q ' \n\n" );
    Asr _ RecStart( hLine ,RAW _ LINEAR _ PCM _ 16BIT) ;
#else
    printf( "推出请按' q ' \n\n" );
#endif
break;
}
return 0L;
}

main( )
{
    cout << " Begin " << endl;
    Asr _ SetSystemDir( ". /Mandarin _ 8KHz3. 0 " );
    int nRet = Asr _ SysInit( );
    if( nRet )
    {
        cout << " init error " << endl;
        return 1;
    }
else
    cout << " init success " << endl;

//生成一个识别引擎
hLine = Asr _ LineInit( ResultProc );
if( ! hLine )
{
    cout << " Engine Create Failed " << endl;
    return 1;
}
else
    cout << " engine create success " << endl;

nRet = Asr _ GrammarLoad( hLine , " test. grm " ,CHARACTER _ GB , " test. lec " );

if( nRet )
{
}

```

```

    cout << " gramload failed " << endl;
    return 1;
}
else
    cout << " gramload success " << endl;

//激活语法规则
nRet = Asr_GrammarActivate ( hLine," talk ");
if( nRet)
{
    cout << " Grammar Activate Failed " << endl;
    Asr_LineRelease( hLine );
    Asr_SysRelease ( );
    return 1;
}
else
    cout << " grammar activate success " << endl;

Asr_SetLog( hLine ,false ) ;
#endif ONLINE _ TEST
Asr_RecordOpen( ); //打开录音设备,
Asr_RecStart( hLine ,RAW _ LINEAR _ PCM _ 16BIT); //开始识别
Asr_RecordStart( cb ); //开始录音
#endif

while( 1 )
{
    Sleep( 100 );
    char command[ 2 ];
    scanf( "% s ",command );
    if( strcmp( _ strlwr( command )," q ") ==0 )
        break;
}

#endif ONLINE _ TEST
Asr_RecordStop( ); //停止录音
Asr_RecordClose( ); //关闭录音设备
#endif

Asr_LineRelease( hLine );

```

```

Asr_SysRelease();
return 0;
}

```

2) 编写语法规则。新建一个记事本命名为 test.grm，在里面写入：

```

$ talk = ( * * * |... ) <0-> $ talks( * * * |... ) <0->;
$ talks =
你好！
你会唱歌吗！
你多大了；

```

注：test.grm 文件增加语法的规则：若“\$ talks =”后面增加要识别的语句，则需另起一行并且顶行开始以“|”结尾，每一条语句为一行，文件的最后一条语句以“；”结尾。否则不能进行识别！

然后再新建一个记事本命名为 test.lec，test.lec 是注音文件，其中第一行的数字是要注音的语句的个数。在里面写入：

```

3
你好 ni3 hao3
你会唱歌吗 ni3 hui4 chang4 ge1 ma1
你多大了 ni3 duo2 da4 le1

```

3) 把 Mandarin_8KHz3.0、asrapi.dll、ASRAPI.h、asrapi.lib (D:\.....\语音控制实验\所需复制的文件\bin 中)、test.grm、test.lec 文件复制到当前程序所在的目录下。

4) 打开 VC++6.0 建立的语音识别控制台程序，设置 link 库文件：打开工程 (project) 菜单 -> setting… 打开工程设置 (project setting) 对话框，选择 link 选项卡，在 object/library modules 编辑框中添加 asrapi.lib (注：如果有多个 link 库文件，文件之间要用空格隔开)，单击 OK。

5) 在工程中加入 ASRAPI.h 文件：单击工作区中的 file view 选项卡，右击 Header Files -> Add files to folder… 打开选择文件对话框选中 ASRAPI.h 文件，单击确定。

6) 最后对程序进行编译和运行，就会出现如图 7-135 的界面，然后对着传声器说你好、你会唱歌吗，你多大了，即可以识别，若要增加识别语句，可以在遵循语法规则在 test.grm 文件和 test.lec 文件里增加。

五、思考题

1. 简述语音识别的原理，并想象语音识别在现实中可应用在哪些方面？

答：语音识别分为预处理、特征提取、模式匹配三部分。预处理过程把声音转换成相应的波形，特征提取模块提取波形的波长，频率等特征，根据提取的特征和模型库中的模型进行匹配，得到相应的文字，这样就完成了整个语音识别的过程。语音识别适合于开发计算机、电话、嵌入式终端等各种设备上的语音应用程序。

2. 利用编好的语音识别模块，使之能够和使用者进行简单的语音互动，如编写一个记事本程序用语音控制字符的输出等。(选做)

如图 7-136 所示为语音识别模块 VC 控制平台运行结果。

```

D:\祁宏燕\语音安装\Yysb\Debug\Yysb.exe

Begin
init success
engine create success
gramload success
grammar activate success

```

图 7-136 语音识别模块 VC 控制平台程序运行结果

语音识别模块函数详细说明：

Asr _ SetSystemDir

- 函数原型：

ASRERROR ASR _ API Asr _ SetSystemDir(const char * psDir) ;

- 输入参数：

psDir：声学模型所在的目录。

- 输出参数：

无。

- 返回值：

返回错误码，ASRERROR _ OK 表明操作成功。

- 描述：

设置识别引擎工作的系统目录，主要是指声学模型所在的目录。在调用初始化函数 Asr _ SysInit 之前应该先调用此函数。

Asr _ SysInit ()

- 函数原型：

ASRERROR ASR _ API Asr _ SysInit(const char * sInCfgFileName = NULL) ;

- 输入参数：

sInCfgFileName：资源配置文件名，默認為 NULL。

- 输出参数：

无。

- 返回值：
返回错误码， ASRERROR _ OK 表明操作成功。
- 描述：
识别系统的初始化函数。

Asr _ LineInit (ResultProc)

- 函数原型：
`HASRLINE ASR _ API Asr _ LineInit(FNRESULTNOTIFY fnNotify, DWORD dwUserData = NULL);`
- 输入参数：
`fnNotify`: 回调函数，识别结果等信息通过此函数返回给用户，详见结构说明。
`DwUserData`: 用户自定义数据。
- 输出参数：
无
- 返回值：
引擎句柄，详见结构说明
- 描述：
创建一个语音识别引擎，并指定该引擎识别结果出来时的回调函数以及用户定义数据。

Asr _ GrammarLoad ()

- 函数原型：
`ASRERROR ASR _ API Asr _ GrammarLoad(HASRLINE hAsrLine, const char * sInGrammarFileName, int nCode = CHARACTER _ GB const char * szLexiconName = NULL);`
- 输入参数：
`hAsrLine`: 识别引擎的句柄，由引擎初始化函数 `Asr _ LineInit` 得到。
`sInGrammarFileName`: 语法规则文件名。
`nCode`: 字符编码，默认为 GB2312。
`szLexiconName`: 注音文件名。
- 输出参数：
无。
- 返回值：
返回错误码， ASRERROR _ OK 表明操作成功。
- 描述：
导入语法规则文件中的语法规则。如果语法中有非标准发音字符或者多音字，可以在第二个参数中提供注音文件，这个文件是文本文件，第一行为符号个数，以后每行为一个词或者符号，然后是空格，后面接发音，多个发音之间用“|”分隔。当这个参数为空时，将尝试查找和语法规则文件名相同，后缀为“lec”的文件。注音规则是：如果注音文件里提

供，直接采用注音文件里的发音，否则尝试自动注音，自动注音失败将返回错误。

Asr_GrammarActivate

- 函数原型：

```
ASRERROR ASR_API Asr_GrammarActivate( HASRLINE hInAsrLine, const
char * pGrammar);
```

- 输入参数：

hAsrLine：识别引擎的句柄，由引擎初始化函数得到。

pGrammar：语法规则名。

- 输出参数：

无。

- 返回值：

返回错误码，ASRERROR_OK 表明操作成功。

- 描述：

设置识别引擎的语法规则。必须首先使用 Asr_GrammarLoad 或 Asr_GrammarLoadFromBuffer 函数载入到系统中。

Asr_RecStart

- 函数原型：

```
ASRERROR ASR_API Asr_RecStart( HASRLINE hInAsrLine, int nWave-
Format);
```

- 输入参数：

hAsrLine：识别引擎的句柄，由引擎初始化函数得到。

CWaveFormat：待识别语音的格式，见头文件定义。

- 输出参数：

无。

- 返回值：

返回错误码，ASRERROR_OK 表明操作成功。

- 描述：

在这一识别引擎中开始等待填充语音数据进行识别。调用此函数必须编写程序采集语音数据，接着调用 Asr_RecFillWaveBuffer 填充语音数据。当数据足够产生结果的时候引擎将会自动结束并返回结果。如果数据已经全部输入还没有得到结果，或者想立即得到结果，可以调用 Asr_RecStop 主动结束并返回结果。

Asr_SetParam

- 函数原型：

```
ASRERROR ASR_API Asr_SetParam ( HASRLINE hInAsrLine , int type,
DWORD data);
```

- 输入参数：

Type：设置参数的类型。

Data: 参数的值。

- **输出参数:**

无。

- **返回值:**

返回错误码，ASRERROR_OK 表明操作成功。

- **描述:**

设置录音以及引擎相关的参数，type 取值以及含义如下：

5：是否连续录音识别。如果为 1 表示连续的录音和反馈识别结果，中间不需要干预。如果为 0 表示每次识别出一个结果后将停止录音，需要再次调用 Asr_RecStartX 才能进行下一次识别。

8：设置在录音的时候是否检测前端点，如果不进行前端点检测，将从调用 Asr_RecStartX 之后，认为所有的声音都为说话声，录音数据送入识别引擎进行识别。

Data 不为 0 表示不进行端点检测，0 为端点检测。默认为进行端点检测。

9：是否进行尾端点检测。在检测前端点的前提下，可以进一步决定是否进行尾端点检测。如果进行尾端点检测，在开始说话后停顿一定的时间就认为该句话结束，停顿时间由本函数 type 为 13 设定。

10：起始语音音量，默认为 100。

11：说话结束音量，默认为 300。

12：起始语音长度，如果超过这个时间，认为说话开始，单位为毫秒，默认为 120。

13：词之间最长停顿时间，如果超过这个时间，将视为说话结束，单位为毫秒，默认为 360。

15：设置录音长度，一旦录音超出这个时间，将强制结束并识别，单位为毫秒。

Asr_RecStart

- **函数原型:**

```
ASRERROR ASR _ API Asr _ RecStart( HASRLINE hInAsrLine, int nWaveFormat);
```

- **输入参数:**

hAsrLine：识别引擎的句柄，由引擎初始化函数得到。

CWaveFormat：待识别语音的格式，见头文件定义。

- **输出参数:**

无。

- **返回值:**

返回错误码，ASRERROR_OK 表明操作成功。

- **描述:**

在这一识别引擎中开始等待填充语音数据进行识别。调用此函数必须编写程序采集语音数据，接着调用 Asr_RecFillWaveBuffer 填充语音数据。

当数据足够产生结果的时候引擎将会自动结束并返回结果。如果数据已经全部输入还没有得到结果，或者想立即得到结果，可以调用 Asr _ RecStop 主动结束并返回结果。

Asr _ GetNBestResult ()

- 函数原型：

HASRLINE hAsrLine , int& nCount , char * * & result , int * & Confidence

- 输入参数：

hAsrLine：识别引擎的句柄，由引擎初始化函数得到。

nCount：根据打分由高到低能够获取的识别结果的个数。

Result：识别结果字符串指针。由 result [0] 到 result [nCount] 依次为最好的结果，次好的结果等。

Confidence：识别结果打分。由 result [0] 到 result [nCount] 依次为最高分，次高分。

- 输出参数：

无。

- 返回值：

返回错误码， ASRERROR _ OK 表明操作成功。

- 描述：

获取按照准确度排序的识别结果以及打分。参数内存分配和释放由 sdk 内部维护。

Asr _ RecStop

- 函数原型：

ASRERROR ASR _ API Asr _ RecStop(HASRLINE hInAsrLine) ;

- 输入参数：

hAsrLine：识别引擎的句柄，由引擎初始化函数得到。

- 输出参数：

无。

- 返回值：

返回错误码， ASRERROR _ OK 表明操作成功。

- 描述：

停止这一个引擎的识别任务，可以在 Asr _ RecStart 函数执行后任何时候执行此函数，此函数执行后要再次识别时需重新调用 Asr _ RecStart 函数。

Asr _ LineRelease (hLine)

- 函数原型：

ASRERROR Asr _ LineRelease (HASRLINE hAsrLine)

- 输入参数：

hAsrLine：识别引擎的句柄，由引擎初始化函数 Asr _ LineInit 得到。

- 输出参数：

无。

- 返回值：
返回错误码， ASRERROR _ OK 表明操作成功。
- 描述：
关闭这一个语音识别引擎，并且释放其相关的资源。

Asr _ SysRelease ()

- 函数原型：
`ASRERROR ASR _ API Asr _ SysRelease();`
- 输入参数：
无。
- 输出参数：
无。
- 返回值：
返回错误码， ASRERROR _ OK 表明操作成功。
- 描述：
系统退出时调用，清除识别系统申请的资源。

同系列书推荐

竞赛机器人

机器智能：人工情感

机器智能：人工心理

机器智能：人脸工程



电话服务

服务咨询热线：010-88361066
读者购书热线：010-68326294
010-88379203

网络服务

机工官网：www.cmpbook.com
机工微博：weibo.com/cmp1952
金书网：www.golden-book.com
教育服务网：www.cmpedu.com
封面无防伪标均为盗版

机工微信

E世界

科技电眼

上架指导 机器人

ISBN:978-7-111-57208-4

策划编辑◎罗莉/封面设计◎MX DESIGN STUDIO Q:1765628429

ISBN 978-7-111-57208-4



9 787111 572084 >

定价：69.00元